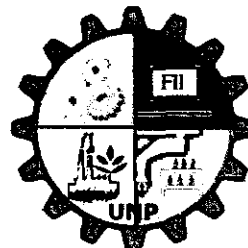




UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE INGENIERÍA INDUSTRIAL
ESCUELA PROFESIONAL DE INGENIERÍA
INFORMÁTICA



TESIS

**Aplicación de nuevas Tecnologías de Información y
Comunicación en la elaboración de un Sistema de Control de
Acreencias a nivel de prototipo para la Entidad Prestadora de
Servicios de Saneamiento Grau y evaluación de su impacto en el
sistema.**

PRESENTADO POR:

Bach. CARLOS AUGUSTO NOLE MACHACA

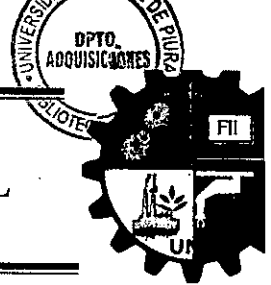
**TESIS PARA OPTAR EL TÍTULO DE
INGENIERO INFORMÁTICO**

PIURA, PERU

2016.



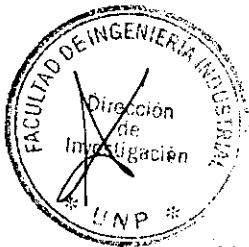
UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE INGENIERÍA INDUSTRIAL
DECANATO



7764
NOL

ACTA DE SUSTENTACIÓN DE TESIS

Los Miembros del Jurado Calificador de la Tesis denominada: «**APLICACIÓN DE NUEVAS TÉCNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN EN LA ELABORACIÓN DE UN SISTEMA DE CONTROL DE ACRENCIAS A NIVEL DE PROTOTIPO PARA LA ENTIDAD PRESTADORA DE SERVICIOS DE SANEAMIENTO GRAU Y EVALUACIÓN DE SU IMPACTO EN EL SISTEMA**», presentado por el señor **NOLE MACHACA CARLOS AUGUSTO**, Bachiller de la Escuela Profesional en **Ingeniería Informática** asesorado por la ING. **QUITO RODRÍGUEZ CARMEN ZULEMA**, Reunidos para la sustentación de ésta y luego de escuchar su exposición y las respuestas a las preguntas formuladas, la declaran:



Con el Calificativo:

APROBADO

BUENO

En consecuencia el sustentante se encuentra **apto** para recibir el título profesional de **INGENIERO INFORMÁTICO** conforme a Ley.

Piura, 19 de Febrero del 2016


Mg. PEDRO CRIOLLO GONZALES
PRESIDENTE - JURADO CALIFICADOR


MSc. JORGE SANDOVAL RIVERA
VOCA - JURADO CALIFICADOR

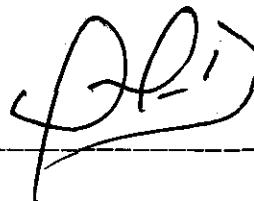

MSc. HECTOR FIESTAS BANCAYAN
SECRETARIO - JURADO CALIFICADOR



Tesis presentada como requisito para optar el título de Ingeniero Informático

ASESOR:

Ing. Arturo Sandoval Rivera



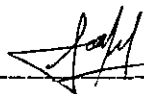
CO - ASESOR:

Ing. Carmen Zulema Quito Rodríguez MSc.



TESISTA:

Bach. Carlos Augusto Nole Machaca



DEDICATORIA

El presente proyecto de investigación está
dedicado a todas aquellas personas que
mediante el uso de las TIC's se han propuesto
mejorar el mundo en que vivimos.

AGRADECIMIENTO

En primer lugar a Dios que me ha brindado la sabiduría,
perseverancia y salud para culminar este proyecto,
a mis padres que sin su apoyo no hubiese sido posible
seguir una carrera profesional, a mis asesores, profesores
y amigos de la universidad, compañeros de trabajo
porque cada uno de ellos me brindaron su experiencia y
conocimientos.

**Aplicación de nuevas Tecnologías de Información y Comunicación
en la elaboración de un Sistema de Control de Acreencias a nivel de
prototipo para la Entidad Prestadora de Servicios de Saneamiento Grau
y evaluación de su impacto en el sistema.**

Carlos Augusto Nole Machaca

RESUMEN

El presente trabajo de investigación tiene como objetivo general analizar y diseñar un prototipo del sistema de control de acreencias que permita gestionar el procedimiento concursal de la Entidad Prestadora de Servicios de Saneamiento Grau S.A. dentro del proceso de reestructuración patrimonial en la cual está inmersa.

Para la realización del SICA se basó en el empleo del marco de trabajo Scrum para la planificación y gestión del proyecto permitiendo que en cada iteración que se finalizaba se obtenía un artefacto funcional el cual iba incrementado. Por el lado de la arquitectura de software se utilizó Spring que convierte a SICA en un sistema robusta, escalable y de fácil mantenimiento.

Las pruebas funcionales y de rendimiento tomando como referencia la ISO / IEC 2500 demostraron que SICA tiene una buena performance y que nos asegura que podrá administrar grandes volúmenes de información que el proceso concursal de la EPS Grau genera.

Finalmente los resultados obtenidos fueron realmente satisfactorios en el cual se demuestra que el SICA aplicando las TIC's de manera adecuada mejora enormemente la gestión del proceso concursal de la EPS Grau.

Application of new Information and Communication Technologies in the development of a Control System DEBTS level prototype for the Lending Institution of Sanitation Services Grau and evaluation of their impact on the system.

Carlos Augusto Nole Machaca

ABSTRACT

The present research work has as general objective to analyze and design a prototype control of credits system that allows to manage bankruptcy procedure of the provider of Sanitation Services Grau SA within the equity restructuring process in which it is immersed and to achieve this objective the following specific objectives were posed:

For the realization of SICA it is based on the use of Scrum framework for planning work and project management allowing each iteration that ended a functional device which was increased was obtained. On the side of the software architecture used SICA Spring that makes a robust, scalable and easy to maintain system.

Functional and performance testing by reference to the ISO / IEC 2500 showed that SICA has a good performance and assures us that can manage large volumes of information that the bankruptcy process generates EPS Grau.

Finally the results were really satisfactory in which it is shown that the SICA applying TIC's properly greatly improves the management of the bankruptcy process of EPS Grau.

ÍNDICE GENERAL

DEDICATORIA.....	i
AGRADECIMIENTO.....	ii
RESUMEN.....	iii
Application of new Information and Communication Technologies in the development of a Control System DEBTS level prototype for the Lending Institution of Sanitation Services Grau and evaluation of their impact on the system.	iv
ABSTRACT.....	iv
ÍNDICE DE CUADROS.....	x
ÍNDICE DE FIGURAS.....	xi
INTRODUCCIÓN.....	1
CAPÍTULO 1: EL PROBLEMA DE LA INVESTIGACIÓN.....	2
1.1. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA.....	2
1.1.1. DESCRIPCIÓN DEL PROBLEMA.....	2
1.1.2. FORMULACIÓN DEL PROBLEMA.....	3
1.2. OBJETIVOS DE LA INVESTIGACIÓN.....	4
1.2.1. OBJETIVO GENERAL.....	4
1.2.2. OBJETIVOS ESPECÍFICOS.....	4
1.2.2.1. DE LA INVESTIGACIÓN.....	4
1.2.2.2. DEL SISTEMA.....	4
1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN.....	4
1.4. HIPÓTESIS DE LA INVESTIGACIÓN.....	5
CAPÍTULO 2: MARCO TEÓRICO.....	6
2.1. MARCO REFERENCIAL.....	6
2.1.1. ANTECEDENTES HISTÓRICOS.....	6
2.1.2. ACTIVIDAD EMPRESARIAL.....	8
2.1.3. FILOSOFÍA EMPRESARIAL.....	9
2.1.3.1. VISIÓN.....	9
2.1.3.2. MISIÓN.....	9
2.1.3.3. VALORES.....	9
2.1.4. PROCESO CONCURSAL.....	9
2.2. MARCO LEGAL.....	11
2.2.1. LEY GENERAL DEL SISTEMA CONCURSAL: 27809.....	11
CAPÍTULO 3: ANÁLISIS DEL SISTEMA.....	14
3.1. METODOLOGÍA.....	14
3.2. PILA DE PRODUCTO (PRODUCT BACKLOG).....	16
3.3. SPRINT 1.....	18

3.3.1. PILA DE SPRINT	18
3.4. SPRINT 2.....	18
3.4.1. PILA DE SPRINT	18
3.5. SPRINT 3.....	19
3.5.1. PILA DE SPRINT	19
3.6. SPRINT 4.....	19
3.6.1. PILA DE SPRINT	19
3.7. SPRINT 5.....	20
3.7.1. PILA DE SPRINT	20
CAPÍTULO 4: DISEÑO DE SISTEMAS	21
4.1. ARQUITECTURA DE LA SOLUCIÓN	21
4.2. DISEÑO DE BASE DE DATOS.....	23
4.2.1. DICCIONARIO DE DATOS	23
TABLA: acreedor.....	23
TABLA: acreencia.....	23
TABLA: asientocontable.....	24
TABLA: asientocontabledetalle	25
TABLA: beneficiario.....	26
TABLA: beneficiario_acreedor.....	27
TABLA: deudaconcurzal.....	28
TABLA: pago.....	29
TABLA: parametros_generales.....	30
TABLA: programacionpago.....	31
TABLA: proveedor	32
TABLA: tipoacreencia	33
TABLA: tipodeuda.....	33
TABLA: tipodocumento.....	34
TABLA: tipoestado	35
TABLA: tipopago.....	35
TABLA: trabajador.....	36
TABLA: users	37
TABLA: user_rols	38
4.2.2. MODELADO DE DATOS.....	39
4.3. DISEÑO DE LA INTERFAZ GRÁFICA	40
4.3.1. AUTENTICACIÓN.	40
4.3.2. MENU PRINCIPAL.....	40

4.2.2.1.	ROL ADMINISTRADOR	41
4.2.2.2.	ROL USUARIO ESTANDAR	41
4.3.3.	MENÚ: PARAMETRÍA	41
4.2.3.1.	PARÁMETRO GENERALES	42
4.2.3.2.	USUARIO	44
4.2.3.3.	TIPO DOCUMENTO	45
4.2.3.4.	TIPO PAGO	45
4.2.3.5.	TIPO DEUDA	46
4.2.3.6.	TIPO ACREENCIA	46
4.2.3.7.	TIPO ESTADO DE ACREENCIA	47
4.3.4.	MENÚ: ACCIONES ESPECIALES	47
4.2.4.1.	CAMBIAR TIPO ACREENCIA	48
4.2.4.2.	EXTORNAR PAGOS	48
4.3.5.	MENU: REGISTRAR	49
4.2.5.1.	ACREENCIA	49
4.2.5.2.	ACREEDOR	50
4.2.4.2.1.	TRABAJADOR	50
4.2.4.2.2.	PROVEEDOR	51
4.2.5.3.	DEUDA CONCURSAL	52
4.2.5.4.	BENEFICIARIOS	52
4.2.5.5.	PROGRAMACIÓN DE PAGOS	53
4.2.5.6.	REPROGRAMACIÓN DE PAGOS	53
4.2.5.7.	PAGOS REALIZADOS SEMESTRAL	54
4.3.6.	MENU REPORTE	56
4.2.5.1.	DEUDA CONCURSAL	56
4.2.5.2.	PAGOS REALIZADOS SEMESTRAL	58
4.2.5.3.	PAGOS EXTORNADOS	60
4.2.5.4.	PROGRAMACIÓN DE PAGOS	60
4.2.5.5.	ACREENCIAS	63
4.2.5.6.	ACREEDOR	64
4.2.5.4.1.	TRABAJADOR	64
4.2.5.4.2.	PROVEEDOR	64
4.2.5.7.	BENEFICIARIO	65
CAPÍTULO 5:	CONSTRUCCIÓN Y PRUEBAS	66
5.1.	FRAMEWORKS	66
5.1.1.	SPRING FRAMEWORK	66

5.1.2.	BOOTSTRAP	67
5.1.3.	SPRING SECURITY	67
5.2.	HERRAMIENTAS DE SOFTWARE	67
5.2.1.	JAVA	67
5.2.2.	HTML	68
5.2.3.	JAVASCRIPT	68
5.2.4.	CSS	69
5.2.5.	iReport Designer	69
5.2.6.	Maven	69
5.2.7.	JSTL	69
5.2.8.	Netbeans	70
5.2.9.	Log4j.....	70
5.3.	PATRONES DE DISEÑO.....	71
5.3.1.	MODELO – VISTA – CONTROLADOR.....	72
5.3.2.	INYECCIÓN DE DEPENDENCIAS.....	73
5.4.	PRUEBAS DEL SICA.....	73
5.4.1.	PRUEBAS DE FUNCIONALIDAD	74
5.4.2.	PRUEBAS DE RENDIMIENTO.....	76
5.4.4.1.	PROGRAMACIÓN DE PAGOS.....	78
5.4.4.2.	REPROGRAMACIÓN DE PAGOS.....	79
5.4.4.3.	PROCESO DE PAGOS.....	80
CAPÍTULO 6:	PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS	81
6.1.	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	81
6.1.1.	PROGRAMACIÓN DE PAGOS.....	81
6.1.1.1.	RESULTADOS EN TABLA	81
6.1.1.2.	REPORTE RESUMEN	85
6.1.2.	REPROGRAMACION DE PAGOS	85
6.1.2.1.	RESULTADOS EN TABLA.....	85
6.1.2.2.	REPORTE RESUMEN	89
6.1.3.	PROCESO DE PAGOS.....	90
6.1.3.1.	RESULTADOS EN TABLA.....	90
6.1.3.2.	REPORTE RESUMEN	94
6.1.4.	INDICADOR 01: TIEMPO DE GENERACIÓN EN REPORTES	94
6.1.5.	INDICADOR 02: NÚMERO DE ÓRDENES DE PAGOS DUPLICADAS	94
6.1.6.	INDICADOR 03: GRADO DE SATISFACCIÓN	95
6.2.	ANÁLISIS COSTO – BENEFICIO DEL SISTEMA	96

6.2.1.	BENEFICIOS TANGIBLES	96
6.2.2.	BENEFICIOS INTAGIBLES	96
6.2.3.	COMPARACION CON EL SICA	96
	CONCLUSIONES.....	98
	RECOMENDACIONES	99
	BIBLIOGRAFÍA	100
	ANEXOS.....	102
ANEXO 1.	DEUDA DE CARÁCTER CONCURSAL O ESTRUCTURAL DE DIFERENTE NATURALEZA	102
ANEXO 2.	registrarAcreencia.jsp.....	103
ANEXO 3.	ENTIDAD ACREEDOR (Acreedor.java)	107
ANEXO 4.	AcreedorDAO.java.....	111
ANEXO 5.	AcreedorBusiness.java	114
ANEXO 6.	AcreedorService.java	116
ANEXO 7.	AcreedorController.java.....	117
ANEXO 8.	AcreedorDTO.java	119
ANEXO 9.	pom.xml.....	120
ANEXO 10.	dispatcher-servlet.xml	130
ANEXO 11.	aplicationContext.xml	131
ANEXO 12.	spring-security.xml.....	132
ANEXO 13.	web.xml	133
ANEXO 14.	jdbc.properties	135
ANEXO 15.	Log4j.properties	135
ANEXO 16.	ProgramacionPagoController.java	135
ANEXO 17.	PagoController.java.....	138
ANEXO 18.	GUÍA DE OBSERVACIÓN NRO. 1 (INDICADOR 01).....	142
ANEXO 19.	GUÍA DE OBSERVACIÓN NRO. 2 (INDICADOR 02).....	143
ANEXO 20.	ENCUESTA NRO. 1 (INDICADOR 03 – NIVEL OPERACIONAL).....	144
ANEXO 21.	ENCUESTA NRO. 2 (INDICADOR 03 – NIVEL ESTRATÉGICO).....	145
ANEXO 22.	DIAGRAMA DE CASOS DE USO DEL SISTEMA	147
ANEXO 23.	DIAGRAMA DE CASOS DE USO DEL NEGOCIO	148
ANEXO 24.	DIAGRAMA DE CLASES	149
	GLOSARIO DE TÉRMINOS	151

ÍNDICE DE CUADROS

Tabla 1. Product Backlog – SICA	17
Tabla 2. Pila de Sprint 1	18
Tabla 3. Pila de Sprint 2	18
Tabla 4. Pila de Sprint 3	19
Tabla 5. Pila de Sprint 4	19
Tabla 6. Pila de Sprint 5	20
Tabla 7. Estructura de la Tabla acreedor	23
Tabla 8. Indices de la Tabla acreedor	23
Tabla 9. Estructura de la Tabla acreencia	24
Tabla 10. Indices de la Tabla acreencia	24
Tabla 11. Estructura de la Tabla asientocontable	25
Tabla 12. Indices de la Tabla asientocontable	25
Tabla 13. Estructura de la Tabla asientocontabledetalle	26
Tabla 14. Indices de la Tabla asientocontabledetalle	26
Tabla 15. Estructura de la Tabla beneficiario	27
Tabla 16. Indices de la Tabla beneficiario	27
Tabla 17. Estructura de la Tabla beneficiario_acreedor	27
Tabla 18. Indices de la Tabla beneficiario_acreedor	28
Tabla 19. Estructura de la Tabla deudaconcurasal	28
Tabla 20. Indices de la Tabla deudaconcurasal	29
Tabla 21. Estructura de la Tabla pago	29
Tabla 22. Indices de la Tabla pago	30
Tabla 23. Estructura de la Tabla parametros_generales	30
Tabla 24. Indices de la Tabla parametros_generales	31
Tabla 25. Estructura de la Tabla programacionpago	31
Tabla 26. Indices de la Tabla programacionpago	32
Tabla 27. Estructura de la Tabla proveedor	32
Tabla 28. Indices de la Tabla proveedor	32
Tabla 29. Estructura de la Tabla tipoacreencia	33
Tabla 30. Indices de la Tabla tipoacreencia	33
Tabla 31. Estructura de la Tabla tipodeuda	34
Tabla 32. Indices de la Tabla tipodeuda	34
Tabla 33. Estructura de la Tabla tipodocumento	34
Tabla 34. Indices de la Tabla tipodocumento	34
Tabla 35. Estructura de la Tabla tipoestado	35
Tabla 36. Indices de la Tabla tipoestado	35
Tabla 37. Estructura de la Tabla tipopago	35
Tabla 38. Indices de la Tabla tipopago	36
Tabla 39. Estructura de la Tabla trabajador	36
Tabla 40. Indices de la Tabla trabajador	37
Tabla 41. Estructura de la Tabla users	37
Tabla 42. Indices de la Tabla users	37
Tabla 43. Estructura de la Tabla user_rols	38
Tabla 44. Indices de la Tabla user_rols	38
Tabla 45. Documento de Trabajo Funcionalidad: Programación de Pagos	74
Tabla 46. Documento de Trabajo Funcionalidad: Reprogramación de Pagos	75
Tabla 47. Documento de Trabajo Funcionalidad: Proceso de Pagos	76

Tabla 48. Resultados en Tabla para la Programación de Pagos	85
Tabla 49. Reporte Resumen para la Programacion de Pagos	85
Tabla 50. Resultados en Tabla para la Reprogramación de Pagos	89
Tabla 51. Reporte Resumen para la Reprogramacion de Pagos	89
Tabla 52. Resultados en Tabla para el Proceso de Pagos	93
Tabla 53. Reporte Resumen para el Proceso de Pagos	94
Tabla 54. Cuadro comparativo de Costos	97
Tabla 55. Guía de Observación Nro. 1	142
Tabla 56. Guía de Observacion Nro. 2	143
Tabla 57. Encuesta Nro. 1	145
Tabla 58. Encuesta Nro. 2	146

ÍNDICE DE FIGURAS

Figura 1. Nexus Framework: for scaling and managing large agile projects	14
Figura 2. Visión Global de la Arquitectura	21
Figura 3. The request processing workflow in Spring Web MVC (high level)	22
Figura 4. Diseño de la Base de Datos	39
Figura 5. Interfaz del Logueo	40
Figura 6. Menú para el Usuario Tipo Administrador	41
Figura 7. Menú para el Usuario Tipo Normal	41
Figura 8. Menú de Parametría	42
Figura 9. Interfaz de Mantenimiento de Parámetros Generales	42
Figura 10. Interfaz de Parámetros Generales luego de realizar la acción Listar	43
Figura 11. Interfaz de Parámetros Generales luego de realizar la acción Editar	43
Figura 12. Interfaz de Mantenimiento de un Usuario	44
Figura 13. Interfaz de Usuario luego de hacer click en Nuevo.	44
Figura 14. Interfaz de Mantenimiento de un Tipo de Documento	45
Figura 15. Interfaz de Mantenimiento de un Tipo Pago	45
Figura 16. Interfaz de Mantenimiento de un Tipo Deuda	46
Figura 17. Interfaz de Mantenimiento de un Tipo Acreencia	46
Figura 18. Interfaz de Mantenimiento de un Tipo Estado de Acreencia	47
Figura 19. Menú de Acciones Especiales	48
Figura 20. Interfaz para el Cambio Tipo Acreencia	48
Figura 21. Interfaz para Extorna Pagos Realizados	48
Figura 22. Menú de Registrar	49
Figura 23. Interfaz de Mantenimiento de una Acreencia	50
Figura 24. Interfaz de Mantenimiento de un Trabajador	51
Figura 25. Interfaz de Mantenimiento de un Proveedor	51
Figura 26. Interfaz de Mantenimiento de una Deuda Concursal	52
Figura 27. Interfaz de Mantenimiento de un Beneficiario	52
Figura 28. Interfaz para el Proceso de Programación de Pagos	53
Figura 29. Interfaz para el Proceso de Reprogramación de Pagos	54
Figura 30. Interfaz para el Proceso de Pagos Realizados Semestralmente	55
Figura 31. Interfaz para la búsqueda de beneficiario por Nro. DNI	55
Figura 32. Interfaz para realizar la búsqueda de un beneficiario por Nombre y/o Apellidos	56
Figura 33. Menú de Reportes	56

Figura 34. Interfaz para generar Reporte de Deuda Concursal	57
Figura 35. Reporte de Deuda Concursal en formato PDF	57
Figura 36. Reporte de Deuda Concursal en formato XLS	58
Figura 37. Interfaz para generar el Reporte de Pagos Realizados Semestralmente	58
Figura 38. Reporte de Pagos Realizados Semestralmente en formato PDF	59
Figura 39. Reporte de Pagos Realizados Semestralmente en formato XLS	59
Figura 40. Interfaz para generar el Reporte de Pagos Extornados	60
Figura 41. Interfaz para generar el Reporte de Programación de Pagos	60
Figura 42. Reporte de Programación de Pagos en formato PDF	61
Figura 43. Reporte de Programación de Pagos en formato XLS	62
Figura 44. Interfaz para generar el Reporte de Acreencias	63
Figura 45. Reporte de Acreencia en formato PDF.	63
Figura 46. Reporte de Acreencia en formato XLS.	63
Figura 47. Interfaz para generar el Reporte de Trabajador.....	64
Figura 48. Interfaz para generar el Reporte de Proveedor.....	64
Figura 49. Interfaz para generar el Reporte de Beneficiario	65
Figura 50. Overview of the Spring Framework.....	66
Figura 51. Estructura de Archivos SICA.....	66
Figura 52. Logs del SICA.....	71
Figura 53. Estructura de un archivo log de SICA.....	71
Figura 54. Una típica colaboración entre los componentes de un MVC	72
Figura 55. The DispatcherServlet.....	73
Figura 56. Plan de Pruebas SICa para medir el rendimiento.....	77
Figura 57. Configuración de Hilos	77
Figura 58. Configuración del Gestor de Autorización HTTP	78
Figura 59. Configuración para los valores por defecto de las peticiones HTTP.....	78
Figura 60. Configuración para realizar la petición de Programación de Pagos	79
Figura 61. Configuración para realizar la petición de Reprogramación de Pagos	79
Figura 62. Configuración para realizar la petición de Pagos.....	80
Figura 63. Gráfico de resultados para la Encuesta Nro. 1	95
Figura 64. . Gráfico de resultados para la Encuesta Nro. 2	95
Figura 65. DCU Sistema - SICA	147
Figura 66. DCU Negocio – SICA.....	148
Figura 67. Diagrama de Clases - SICA	150

INTRODUCCIÓN

Hace mucho tiempo, la función de la informática dentro de las empresas se ha considerado por la alta administración como una herramienta para apoyar las funciones operativas. La perspectiva actual y futura tiende a cambiar radicalmente este enfoque. Ahora, los sistemas de información son vistos además como áreas de oportunidad para lograr ventajas en el terreno de los negocios, ya que éstos pueden representar un valor agregado con respecto a sus competidores lo que les permite una diferenciación ante ellos.

El análisis y diseño del prototipo del sistemas de control de acreencias (SICA) permitirá la gestión y el debido control del proceso concursal de la Entidad Prestadora de Servicios de Saneamiento Grau Sociedad Anónima – EPS GRAU S.A teniendo un registro adecuado de las deudas reconocidas y no reconocidas, ofreciendo información pertinente la que se usará en el área administrativa para la adecuada toma de decisiones basada en los diferentes tipos de reportes que SICA les generará.

En el **Capítulo 1** se abordará el problema de la investigación, la descripción del problema, los objetivos de la investigación y del sistema, la justificación e hipótesis de la investigación; luego en el **Capítulo 2** se habla sobre el marco teórico que incluye el referencial y legal donde se expone la información científica disponible sobre nuestra investigación; en el **Capítulo 3** se realiza todo el análisis del sistema que incluye desde la detección de los requerimientos y la metodología a emplear; en el **Capítulo 4** trataremos sobre el diseño de sistemas que abarca la arquitectura del Sistema, el diseño de la base de datos y las interfaces que SICA tendrá; el **Capítulo 5** es donde se trata de los framework, herramientas de software, patrones de diseño que nos permitieron la contrucción del prototipo SICA y las pruebas que se realizaron en ella y finalmente el **Capítulo 6** se presenta y se discuten los resultados obtenidos luego de aplicar lo mencionado en capítulos anteriores.

CAPÍTULO 1: EL PROBLEMA DE LA INVESTIGACIÓN

1.1. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

1.1.1. DESCRIPCIÓN DEL PROBLEMA.

La Entidad Prestadora de Servicios de Saneamiento Grau Sociedad Anónima - EPS GRAU S.A. es una empresa municipal de derecho privado, constituida como sociedad anónima con autonomía administrativa, técnica y económica. Está conformada por cinco municipalidades provinciales y veinte distritales comprendidas dentro del departamento de Piura, donde la Entidad ejerce su jurisdicción.

El objeto social de la EPS GRAU S.A, es la prestación de los servicios de producción y distribución de agua potable; recolección, tratamiento y disposición del alcantarillado sanitario y pluvial y, el servicio de disposición sanitaria de excretas, sistema de letrinas y fosas sépticas en el ámbito de las ciudades de Piura, Sullana, Talara, Paita y Chulucanas – Morropón del Departamento de Piura en la República de Perú.

El 06 de noviembre del 2000, la Comisión de Reestructuración Patrimonial del Instituto Nacional de Defensa de la Competencia y de Protección de la Propiedad Intelectual (INDECOPI) declaró el estado de insolvencia de la EPS GRAU S.A., motivo por el cual ingresó a un proceso concursal, en función de las deudas se otorgó y se eligió la primera junta de acreedores, quienes elaboraron un plan de pagos a 30 años de las deudas concursales el cual fue aprobado por dicha junta. Esto permitió que la empresa continúe desempeñando sus actividades de servicio de saneamiento.

Es proceso concursal generó la clasificación de los créditos según lo señalado en la Ley General del Sistema Concursal comprenden: laborales, comerciales y tributarios, no reconocidos y contingentes. Los créditos en la condición de **reconocidos**, calificación otorgada por INDECOPI le permite al acreedor hacer efectivo el cobro de su crédito de acuerdo a la programación de pagos del plan de Reestructuración

Los créditos **No Reconocidos y Contingentes**, también forman parte de la deuda concursal general, tienen una programación de pagos que se hace efectiva cuando proceda su reconocimiento, para esto la EPS GRAU provisiona a nivel de caja el pago respectivo.

Actualmente tiene que cumplir con los pagos a sus acreedores de todas aquellas acreencias reconocidas y que se encuentra en la programación previamente aprobada. El control es mediante una Hoja de Cálculo de Excel que se vuelve engorroso, sobre todo al momento de generar información de deudas y/o pagos realizado porque se debe realizar manualmente y se tiene una capacidad limitada de respuesta en generar este tipo de reportes, y esto conlleva también a cometer errores para tesorería por ejemplo en el caso cuando a un acreedor se le ha cancelado más veces a pesar que se concluyó con sus pagos.

Como es sabido, la EPS GRAU depende de la toma de decisiones para hacer funcionar su negocio, y las mejores decisiones, a su vez, dependen de contar con información confiable y oportuna por lo cual el gran volumen de información que se produce por el proceso concursal por el que atraviesa ya no es la mejor opción seguir llevándolo de manera manual y con una hoja de cálculo.

Esto amerita que urgentemente se cree un sistema de información computacional para controlar las deudas y los pagos que están programados en el plan de pagos que concluye en el año 2035.

1.1.2. FORMULACIÓN DEL PROBLEMA.

¿Cómo la aplicación de nuevas tecnologías de información y comunicación permitirá gestionar adecuadamente las deudas y los pagos efectuados en el proceso concursal que se encuentra la EPS Grau?

1.2. OBJETIVOS DE LA INVESTIGACIÓN

1.2.1. OBJETIVO GENERAL.

Aplicar nuevas tecnologías de información y comunicación en la elaboración de un prototipo de un sistema de control de acreencias que permitirá gestionar el procedimiento concursal de la Entidad Prestadora de Servicios de Saneamiento Grau S.A. dentro del proceso de reestructuración patrimonial en la cual está inmersa.

1.2.2. OBJETIVOS ESPECÍFICOS.

1.2.2.1. DE LA INVESTIGACIÓN.

- Mejorar el Sistema de control de acreencias automatizando el proceso concursal.
- Mejorar el control de los pagos realizados por la empresa EPS Grau de acuerdo a la programación establecida.
- Disminuir los tiempos de búsqueda y generación de reportes a través de Sistema automatizado.

1.2.2.2. DEL SISTEMA.

- Realizar el análisis y definición de los requerimientos del sistema y negocio del proceso concursal.
- Diseñar el modelado de la Base de Datos del sistema.
- Desarrollar el sistema utilizando la metodología ágil Scrum.
- Implementar el sistema utilizando el framework Spring.
- Determinar los indicadores para medir la eficiencia del sistema.
- Determinar el nivel de satisfacción de los usuarios respecto al sistema implementado.

1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Este sistema pretende llevar una administración, ordenada, eficiente y eficaz del proceso concursal de la Empresa Prestadora de Servicios de Saneamiento Grau S.A. y contar con información oportuna y precisa de la deuda concursal, llevando un registro adecuado de las acreencias y el control de los pagos según la programación aprobada

por la junta de acreedores dentro del plan de reestructuración patrimonial. Esta información podrá ser visualizada y/o reportada en forma oportuna.

SICA ahorrará horas hombre en generar información tanto para la gestión como para los acreedores dado que el proceso de concursal estará automatizado.

1.4. HIPÓTESIS DE LA INVESTIGACIÓN

El sistema SICA mejorará la gestión del proceso concursal de la empresa “EPS GRAU S.A.”

CAPÍTULO 2: MARCO TEÓRICO

2.1. MARCO REFERENCIAL

2.1.1. ANTECEDENTES HISTÓRICOS.

ANTES DE 1980: MINISTERIO DE FOMENTO Y OBRAS PÚBLICAS.-

La administración de los Servicios de Saneamiento de Piura y Tumbes, en la década de los 80 correspondía a la Sub – Dirección de Obras Sanitarias que dependía directamente de la Dirección General de Obras Sanitarias del Ministerio de Fomento y Obras Públicas.

1981: SE CREA SENAPA.- En el año 1981, mediante Decreto Legislativo N° 150, se crea el Servicio Nacional de Abastecimiento de Agua Potable y Alcantarillado – SENAPA, una empresa estatal que reemplazó la Dirección General de Obras Sanitarias – Ministerio de Fomento y Obras Públicas – y absolvió a las empresas autónomas que poseía en Lima, Arequipa y Trujillo.

1981 – 1983: DIRECCION REGIONAL DE VIVIENDA Y CONSTRUCCION.- A partir de 1981 los servicios de Agua Potable y Alcantarillado del Departamento de Piura, pasaron a depender de la Dirección Regional de Vivienda y Construcción, quien a su vez dependía administrativamente del ORDENORTE y normativamente del Ministerio de Vivienda y Construcción.

MEDIADOS DE 1983: SE CREA LA UNIDAD OPERATIVA PIURA – TUMBES.- A mediados de 1983, la Unidad Operativa Piura – Tumbes SENAPA; asume la administración de los Servicios de Saneamiento de los Departamentos de Piura y Tumbes, dependiente del Servicio Nacional de Abastecimiento de Agua Potable y Alcantarillado - SENAPA.

AGOSTO DE 1983: SE CREA SEDAPIURA.- En agosto de 1983, se crea la Empresa Filial de SENAPA denominada: Servicio de Agua Potable y Alcantarillado de Piura – SEDAPIURA, dependiendo normativamente de

SENAPA y su organización y administración estaban a cargo de una Junta Empresarial, un Directorio y la Gerencia General.

ABRIL DE 1990: TRANSFIEREN ACCIONES A CONCEJOS PROVINCIALES Y DISTRITALES.- A partir del 30 de Abril de 1990, por disposición expresa del Decreto Legislativo 574 complementado por el Decreto Legislativo N° 601, SENAPA transfería a Título gratuito las acciones a las Municipalidades Provinciales y Distritales, estableciendo por ende como órganos de Dirección y Administración: la Junta Empresarial, el Directorio y la Gerencia General, teniendo como órgano rector al Ministerio de la Presidencia.

DICIEMBRE 1992 – SE CREA SUNASS.- En diciembre de 1992, se crea, mediante Decreto Ley N°25965, la Superintendencia Nacional de Servicios de Saneamiento (SUNASS) como el organismos encargado de promover el desarrollo y proponer las normas para la prestación de los servicios de saneamiento, fiscalizar la prestación y resolver, en última instancia , los reclamos de los usuarios. Entonces, se declara en disolución y liquidación a la Empresa Servicio Nacional de Abastecimiento de Agua Potable y Alcantarillado SENAPA (Decreto Ley N° 25973).

JULIO 1994: LEY GENERAL DE SERVICIOS DE SANEAMIENTO.- En Julio de 1994, se aprueba la Ley N° 26338, Ley General de Servicios de Saneamiento, reglamentada mediante Decreto Supremo 09 – 95 – PRES, agosto de 1995, que establece las competencias de las instituciones involucradas en el sector, determina los deberes y derechos de los usuarios y las Entidades Prestadoras, define el régimen tarifario, establece las conciliaciones y modalidades de la participación privada, entre otros temas.

MARZO 1995: SE RECONOCE COMO ENTIDAD PRESTADORA A LA EPS GRAU S.A.- Con Resolución de Superintendencia N° 036 – 95 – PRES/VMI/SSS, de fecha 03 de marzo de 1995, la Superintendencia reconoce como Entidad Prestadora de Servicios de Saneamiento a la Empresa de Servicios de Agua Potable de Piura (SEDAPIURA).

1996 – SEDAPIURA CAMBIA DE RAZON SOCIAL POR LA DE EPS GRAU S.A.- En 1996 con Resolución de Superintendencia N° 026 – 96 – PRES/VMI/SUNASS, se aprueba la Directiva sobre adecuación estatutaria de las EPS, a las disposiciones contenidas en la Ley General de Servicios de Saneamiento y su Reglamento. Bajo este contexto y por Acuerdo de Directorio N° 03 – 30 – 96, se aprobó el cambio de razón social quedando como Empresa Prestadora de Servicios de Saneamiento GRAU S.A – EPS GRAU S.A, hasta la fecha.

MARZO – 2000: SE DECLARA EN ESTADO DE INSOLVENCIA A LA EPS GRAU S.A.- El 06 de Noviembre de 2000, la Comisión de Reestructuración Patrimonial del Instituto Nacional de Defensa de la Competencia y Protección de la Propiedad Intelectual - INDECOPI, con Resolución N° 303 – 2000/CRP – PIURA, declaró de oficio el estado de insolvencia de la EPS GRAU S.A.

ABRIL – 2010.- Actualmente la empresa sigue inmersa en el Proceso Concursal.

2.1.2. ACTIVIDAD EMPRESARIAL.

La Entidad Prestadora de Servicios de Saneamiento Grau Sociedad Anónima - EPS GRAU S.A. es una empresa municipal de derecho privado, constituida como sociedad anónima con autonomía administrativa, técnica y económica. Está conformada por cinco municipalidades provinciales y veinte distritales comprendidas dentro del departamento de Piura, donde la Entidad ejerce su jurisdicción.

El objeto social de la EPS GRAU S.A., es la prestación de los servicios de producción y distribución de agua potable; recolección, tratamiento y disposición del alcantarillado sanitario y pluvial y, el servicio de disposición sanitaria de excretas, sistema de letrinas y fosas sépticas en el ámbito de las ciudades de Piura, Sullana, Talara, Paita y Chulucanas – Morropón del Departamento de Piura en la República del Perú.

2.1.3. FILOSOFÍA EMPRESARIAL.

2.1.3.1. VISIÓN.

Ser al 2016, una empresa líder en el sector saneamiento del norte del país, cumpliendo su Plan de Reestructuración se transformará en una entidad moderna y rentable, para satisfacer en términos de calidad, la demanda de los servicios de agua potable y alcantarillado dentro del ámbito de su jurisdicción, en concordancia a políticas de desarrollo sectorial, ordenado y ambientalmente sostenible.

2.1.3.2. MISIÓN.

Administrar eficientemente el recurso agua, así como recolectar y tratar de manera adecuada el alcantarillado, para obtener resultados empresariales óptimos establecidos en el Plan de Reestructuración; con la participación de todos los trabajadores, comprometidos y motivados con el mejoramiento de la calidad de vida de la población y la preservación del medio ambiente.

2.1.3.3. VALORES.

- Respeto
- Honestidad
- Responsabilidad
- Vocación de servicio
- Trabajo en equipo
- Lealtad

2.1.4. PROCESO CONCURSAL.

1. En el mes de marzo del año 2000, la Comisión de Reestructuración Patrimonial de INDECOPI admitió el pedido de la Junta General de Accionistas de la EPS GRAU S.A. de acogerse al Procedimiento Transitorio regulado por el D.U. N° 064-99.

2. El 16 de octubre del año 2000 se instala Junta de Acreedores y se conformó el Comité Transitorio de Créditos, el cual, en acuerdo con los acreedores, decide no aprobar El Plan de Saneamiento presentado por la EPS Grau S.A. Por este motivo, INDECOPI procedió a declarar, de oficio, el estado de insolvencia de dicha empresa mediante la Resolución N° 303-2000/CRP-PIURA fechada el 6 de noviembre del año 2000, la cual fue confirmada con la Resolución N° 110-2001/TDC-INDECOPI, de fecha 16 de febrero del año 2001.
3. El 19 de marzo de 2001, se publicó en El Peruano la declaratoria de insolvencia de la EPS GRAU S.A.
4. Con fecha 12 de julio de 2001, se reunió la Junta de Acreedores de la EPS GRAU S.A., en la que, entre otros asuntos, se decidió la reestructuración patrimonial de dicha empresa.
5. Con fecha 04 de diciembre de 2003 la Comisión de Procedimientos concursales del INDECOPI PIURA convoca a Junta de Acreedores para los días 12 y 15 de enero 2004 (Primera y Segunda Convocatoria respectivamente) para tratar, entre otros temas, la aprobación del Plan de Reestructuración de EPS Grau S.A.
6. El día 15 de enero de 2,004 se aprobó, en segunda convocatoria, el Plan de Reestructuración Patrimonial, el mismo que fue suscrito por el Presidente de la Junta de Acreedores y por el representante de la Administración Mixta.

En marzo del año 2007, la Junta de Acreedores de EPS GRAU S.A, aprobó la Cuarta Modificatoria del Plan de Reestructuración de acuerdo a lo indicado por el Art. 66° de la Ley General del Sistema Concursal Ley 27809, el mismo que considera el pago del 100% de la deuda reconocida entre los años 2 y 24. Asimismo se establece una provisión para la deuda no reconocida en el mismo año 24 y para la contingente entre los años 25 y 30.

Actualmente, nuestra Junta de Acreedores se encuentra compuesta por las autoridades debidamente elegidas de acuerdo al Art. 54° de la Ley General del Sistema Concursal, Ley N° 27809, cuyo presidente es el Ing. Plácido Aguirre Alata representante de Contribuciones Reembolsables del Ministerio de Economía y Finanzas y el Econ. Juan Carlos Barandiarán Rojas, como Vicepresidente y representante de los Créditos Directos del Ministerio de Vivienda y Construcción (FONAVI).

Asimismo, EPS Grau actualmente tiene deudas de carácter concursal o estructural de diferente naturaleza. (**ANEXO 1**). Por otro lado es necesario precisar que EPS GRAU S.A. viene cumpliendo semestralmente con lo establecido en el Plan de Reestructuración Patrimonial respecto a los pagos de la Deuda Concursal de naturaleza laboral.

2.2. MARCO LEGAL

2.2.1. LEY GENERAL DEL SISTEMA CONCURSAL: 27809.

OBJETIVO DEL SISTEMA CONCURSAL.- El objetivo del Sistema Concursal es la permanencia de la unidad productiva, la protección del crédito y el patrimonio de la empresa. Los agentes del mercado procurarán una asignación eficiente de sus recursos durante los procedimientos concursales orientando sus esfuerzos a conseguir el máximo valor del patrimonio en crisis.

FINALIDAD DE LOS PROCEDIMIENTOS CONCURSALES.- Los procedimientos concursales tienen por finalidad propiciar un ambiente idóneo para la negociación entre los acreedores y el deudor sometido a concurso, que les permita llegar a un acuerdo de reestructuración o, en su defecto, a la salida ordenada del mercado, bajo reducidos costos de transacción.

DECISIÓN SOBRE EL DESTINO DEL DEUDOR.- La viabilidad de los deudores en el mercado es definida por los acreedores involucrados en los respectivos procedimientos concursales, quienes asumen la responsabilidad y consecuencias de la decisión adoptada.

UNIVERSALIDAD.- Los procedimientos concursales producen sus efectos sobre la totalidad del patrimonio del deudor, con las excepciones establecidas expresamente por la ley.

COLECTIVIDAD.- Los procedimientos concursales buscan la participación y beneficio de la totalidad de los acreedores involucrados en la crisis del deudor. El interés colectivo de la masa de acreedores se superpone al interés individual de cobro de cada acreedor.

PROPORCIONALIDAD.- Los acreedores participan proporcionalmente en el resultado económico de los procedimientos concursales, ante la imposibilidad del deudor de satisfacer con su patrimonio los créditos existentes, salvo los órdenes de preferencia establecidos expresamente en la presente Ley.

INICIO E IMPULSO DE LOS PROCEDIMIENTOS CONCURSALES.- Los procedimientos concursales se inician a instancia de parte interesada ante la autoridad concursal. El impulso de los procedimientos concursales es de parte. La intervención de la autoridad concursal es subsidiaria.

CONDUCTA PROCESAL.- Los sujetos del procedimiento, sus representantes, sus abogados y, en general, todos los partícipes de los procedimientos concursales, deben adecuar su conducta a los deberes de veracidad, probidad, lealtad y buena fe. La temeridad, mala fe o cualquier otra conducta dolosa son objeto de sanción, de acuerdo a Ley.

INTEGRACIÓN DE LA NORMA.- La autoridad concursal no podrá dejar de resolver por defecto o deficiencia de las normas. En tal caso, aplicará los principios generales del derecho, especialmente aquellos que inspiran el Derecho Concursal.

ROL PROMOTOR DEL ESTADO.- El Estado, a través del INDECOPI, facilita y promueve la negociación entre acreedores y deudores, respetando la autonomía privada respecto de las decisiones adoptadas en los procedimientos concursales con las formalidades de ley.

CAPÍTULO 3: ANÁLISIS DEL SISTEMA

3.1. METODOLOGÍA

El desarrollo del presente proyecto, se basa en **SCRUM** porque ofrece un marco de trabajo para desarrollar proyectos de una forma iterativa e incremental. El desarrollo se estructura en ciclos de trabajo llamado Sprints (también conocidos como iteraciones las cuales no deben durar más de cuatro semanas).

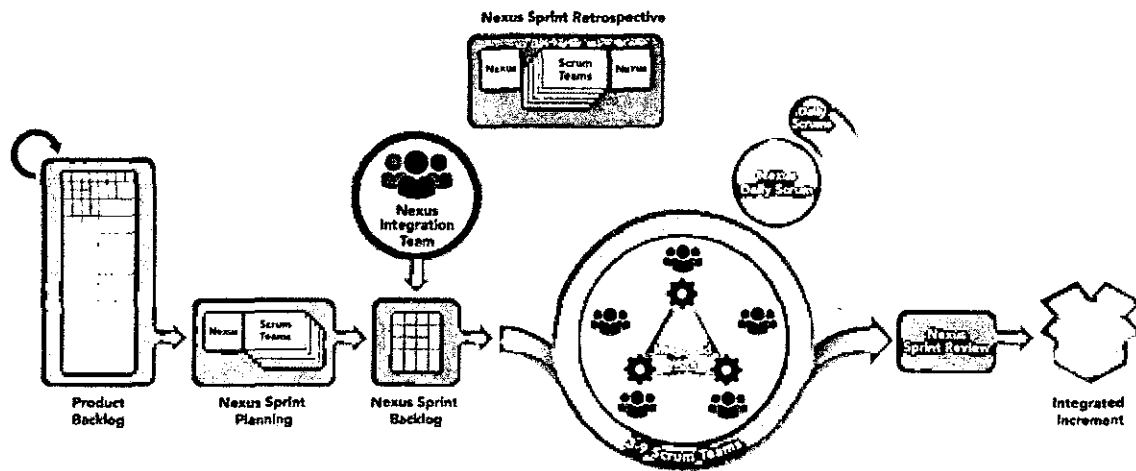


Figura 1. Nexus Framework: for scaling and managing large agile projects
Fuente: <http://www.scrum.org/portals/0/Graphics/Scaling/Nexus%20Scrum.png>

En primer lugar se selecciona **elementos** (peticiones de cliente) de una lista priorizada. Al final del Sprint se entrega algo tangible y que está terminado por completo. Todos los días el Equipo se reúne brevemente para inspeccionar su progreso y ajustar los siguientes pasos necesarios para completar el trabajo pendiente. Al final de cada Sprint, el Equipo revisa el Sprint con los diferentes Stakeholders (interesados e involucrados en el producto) y realiza una demostración de lo que han desarrollado. Se obtiene un feedback que podrá ser incorporado en el siguiente Sprint.

Un lema recurrente en Scrum es “inspección y adaptación” (Kniberg, 2007). Dado que el desarrollo con lleva de forma inevitable aprendizaje, innovación y sorpresas.

Inspección.- Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones no deseables.

Adaptación.- Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados.

Scrum prescribe formalmente cuatro oportunidades para la inspección y adaptación:

- Reunión de Planificación del Sprint (Sprint Planning Meeting)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

(Schawaber & Sutherland, 2011)

En Scrum existen tres roles:

El **Dueño de Producto** es el responsable de maximizar el retorno de la inversión (ROI) a base de identificar las funcionalidades de producto, trasladarlas a una lista priorizada, decidir cuáles deberían estar al principio de la lista para el siguiente Sprint, y repriorizar y refinar continuamente dicha lista.

El **Equipo** (También llamado **Equipo de Desarrollo**) construye lo que el Dueño de Producto indica.

El **ScrumMaster** ayuda al área de producto a aprender y aplicar Scrum para obtener valor de negocio. El ScrumMaster hace todo lo que esté en su mano para ayudar al Equipo, al Dueño de Producto y a la organización a tener éxito. (Deemer, Larman, Benefield, & Vodde, 2012)

3.2. PILA DE PRODUCTO (PRODUCT BACKLOG)

ID	NOMBRE	IMPORTANCIA	ESTIMACION INICIAL	COMO PROBARLO
01	Mantenimiento de Parametría	40	3	1. Se podrá guardar, actualizar la información (usuario, tipo documento, tipo pago, etc.). 2. Al ingresar a un formulario por ejemplo de pago se cargará los valores guardados en Tipo de Pago.
02	Mantenimiento de las acreencias	70	5	Guardar la información de la acreencia o actualizarla.
03	Mantenimiento de Acreedor	55	3	Guardar la información del acreedor o actualizarla.
04	Cambiar tipo de acreencias	60	4	Actualizar información de la acreencia.
05	Mantenimiento Deuda Concursal	80	5	Guardar la información de la deuda concursal o actualizarla.
06	Mantenimiento de beneficiarios	50	3	Guardar la información del beneficiario o actualizarla.
07	Registrar Programación de Pagos	75	4	El sistema le permita generar la programación de pagos.
08	Registrar Reprogramación de pagos	78	4	El sistema le permita generar la reprogramación de pagos.
09	Mantenimiento de pagos realizados semestralmente	76	5	Guardar la información del pago o actualizarla.

10	Reporte de Deuda Concursal	85	5	Luego de terminar el proceso se podrá visualizar el reporte en formato PDF y/o XLS.
11	Reporte de Pagos Realizados Semestralmente	77	5	Luego de terminar el proceso se podrá visualizar el reporte en formato PDF y/o XLS.
12	Reporte de Acreencias	72	5	Luego de terminar el proceso se podrá visualizar el reporte en formato PDF y/o XLS.
13	Reporte de Acreedores	57	4	Luego de terminar el proceso se podrá visualizar el reporte en formato PDF y/o XLS.
14	Reporte de Beneficiarios	56	4	Luego de terminar el proceso se podrá visualizar el reporte en formato PDF y/o XLS.
15	Extornar Pagos	30	3	El sistema permitirá extornar un pago que se haya efectuado.
16	Reporte de Pagos Extornados	30	3	Se podrá visualizar el reporte en formato PDF y/o XLS.

Tabla 1. Product Backlog – SICA

Fuente: Elaborado por el autor.

3.3. SPRINT 1

3.3.1. PILA DE SPRINT

OBJETIVO DE PRINT

Establecer la parametría del SICA, funcionalidad para las Acreencias y Acreedor.

PERIODO DE SPRINT

01/01/2015 – 21/01/2015

ID. HISTORIA

NOMBRE

IMPORTANCIA

ESTIMACIÓN

01	Mantenimiento de Parametría	40	3
02	Mantenimiento de las acreencias	70	5
03	Mantenimiento de Acreedor	55	3
04	Cambiar tipo de acreencias	60	4

Tabla 2. Pila de Sprint 1
Fuente: Elaborado por el autor.

3.4. SPRINT 2

3.4.1. PILA DE SPRINT

OBJETIVO DE PRINT

Funcionalidad para la Deuda Concursal, Beneficiarios y Generar la Programación de Pagos.

PERIODO DE SPRINT

26/01/2015 – 10/02/2015

ID. HISTORIA

NOMBRE

IMPORTANCIA

ESTIMACIÓN

05	Mantenimiento Deuda Concursal	80	5
06	Mantenimiento de beneficiarios	50	3
07	Registrar Programación de Pagos	75	4

Tabla 3. Pila de Sprint 2
Fuente: Elaborado por el Autor.

3.5. SPRINT 3

3.5.1. PILA DE SPRINT

OBJETIVO DE PRINT

Generar la Reprogramación de pagos, Realizar Pagos y Generar Reporte sobre las Deudas Concursales

PERIODO DE SPRINT

13/02/2015 – 04/03/2015

ID. HISTORIA	NOMBRE	IMPORTANCIA	ESTIMACIÓN
08	Registrar Reprogramación de pagos	78	4
09	Mantenimiento de pagos realizados semestralmente	76	5
10	Reporte de Deuda Concursal	85	5

Tabla 4. Pila de Sprint 3
Fuente: Elaborado por el Autor

3.6. SPRINT 4

3.6.1. PILA DE SPRINT

OBJETIVO DE PRINT

Generar la Reportería de los pagos, acreencias y acreedores.

PERIODO DE SPRINT

09/03/2015 – 27/03/2015

ID. HISTORIA	NOMBRE	IMPORTANCIA	ESTIMACIÓN
11	Reporte de Pagos Realizados Semestralmente	77	5
12	Reporte de Acreencias	72	5
13	Reporte de Acreedores	57	4

Tabla 5. Pila de Sprint 4
Fuente: Elaborado por el Autor

3.7. SPRINT 5

3.7.1. PILA DE SPRINT

OBJETIVO DE PRINT	Funcionalidad 100% del SICA.		
PERIODO DE SPRINT	30/03/2015 – 15/04/2015		
ID. HISTORIA	NOMBRE	IMPORTANCIA	ESTIMACIÓN
14	Reporte de Beneficiarios	56	4
15	Extornar Pagos	30	3
16	Reporte de Pagos Extornados	30	3

Tabla 6. Pila de Sprint 5
Fuente: Elaborado por el Autor

CAPÍTULO 4: DISEÑO DE SISTEMAS

4.1. ARQUITECTURA DE LA SOLUCIÓN

Debido a que usuarios de nivel estratégico serán los beneficiarios del sistema se eligió una Arquitectura Web, en la cual los clientes podrán acceder desde cualquier equipo de cómputo a través de su navegador web sin necesidad que se instale algún aplicativo adicional.

El SICA estará alojado en un servidor de aplicaciones, exactamente en el Apache Tomcat. Luego como motor de base de datos se eligió MySQL.

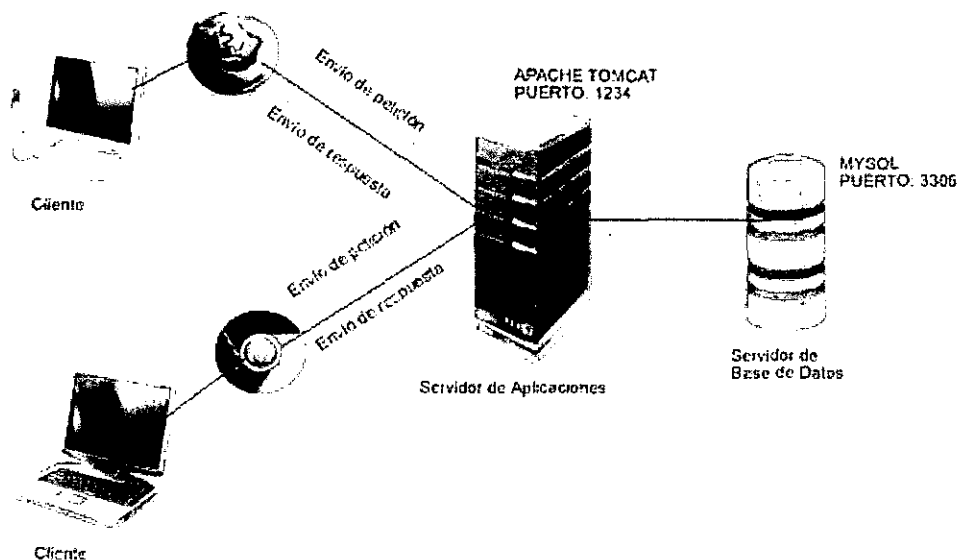


Figura 2. Visión Global de la Arquitectura
Fuente: Elaborado por el Autor

En cuanto la Arquitectura de Software se eligió Spring Web MVC Framework que está diseñado alrededor de un DispatcherServlet que despacha las peticiones a los gestores de eventos (handler) donde cada handler está basado en un controlador que es responsable de preparar el modelo y seleccionar la vista donde se resolverá.

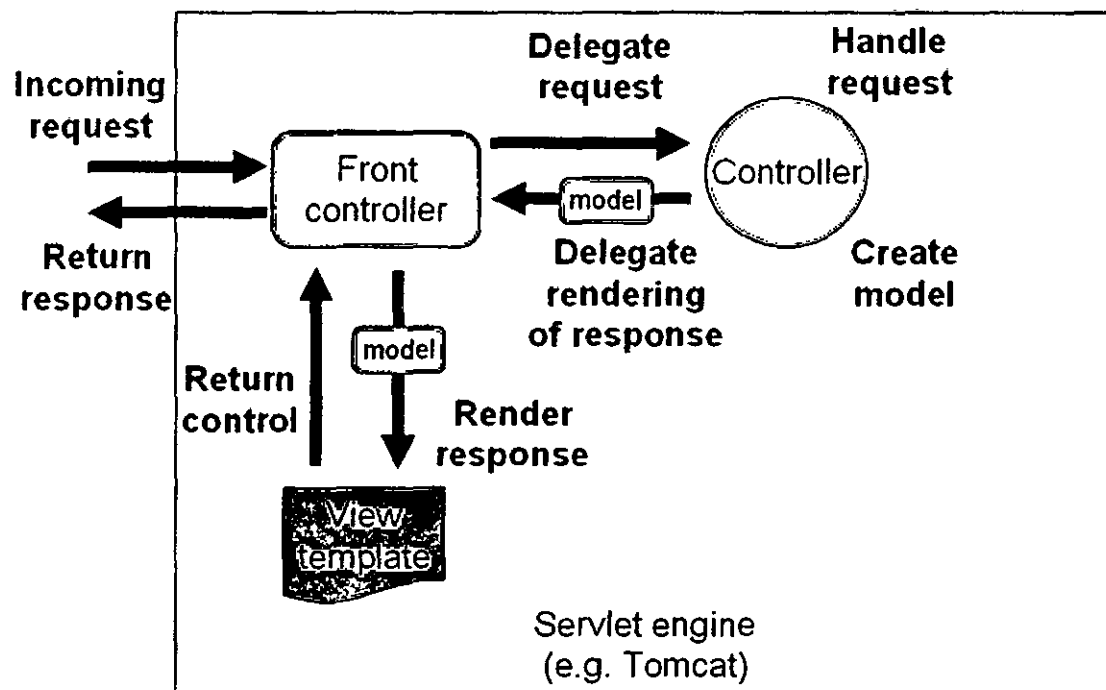


Figura 3. The request processing workflow in Spring Web MVC (high level)

Fuente: <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/mvc.html>

SICA utiliza Java EE (Java Platform, Enterprise Edition) que es la plataforma para el desarrollo de aplicaciones empresariales, que permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Posee varias tecnologías de Servicios Web como JavaBeans, servlets, portlets, JavaServer Pages, etc.

4.2. DISEÑO DE BASE DE DATOS

4.2.1. DICCIONARIO DE DATOS

TABLA: acreedor

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idAcreedor	int(11)	No			Clave primaria, autoincremental.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Sí N=No
tipoAcreedor	int(11)	Sí	NULL		Tipo de Acreedor: 1=Trabajador, 2=Proveedor

Tabla 7. Estructura de la Tabla acreedor
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idAcreedor	8	A		

Tabla 8. Indices de la Tabla acreedor
Fuente: Elaborado por el Autor

TABLA: acreencia

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idAcreencia	int(11)	No			Clave primaria, autoincremental.
num_documento	varchar(15)	Sí	NULL		Numero de documento.
monto_reconocido	Double	Sí	NULL		Monto reconocido a pagar.
Resolución	Longtext	Sí	NULL		Resolución dada por INDECOPI.
fecha_resolucion	Date	Sí	NULL		Fecha en que se emitió la Resolución.

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
fecha_registro	Date	Si	NULL		Fecha en que se registró la Acreencia.
idTipoEstado	int(11)	Si	NULL	tipoestado -> idTipoEstado	Identificador del Tipo de Estado.
idDeudaConcursal	int(11)	Si	NULL	deudaconcursal -> idDeudaConcursal	Identificador de la Deuda Concursal.
idTipoAcreencia	int(11)	No			
idAcreedor	int(11)	Si	NULL		Identificador del Acreedor.
userAdd	varchar(45)	Si	NULL		Usuario que agrego el registro
dateAdd	Datetime	Si	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Si	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Si	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Si	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 9. Estructura de la Tabla acreencia

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Si	No	idAcreencia	2	A		
fk_acre_deucon_idDeudaConcursal	BTREE	No	No	idDeudaConcursal	2	A	YES	
fk_acre_tipest_idTipoEstado	BTREE	No	No	idTipoEstado	2	A	YES	
re_in_idTipAcre	BTREE	No	No	idTipoAcreencia	2	A		
re_in_idAcree	BTREE	No	No	idAcreedor	2	A	YES	

Tabla 10. Indices de la Tabla acreencia

Fuente: Elaborado por el Autor

TABLA: asientocontable

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idAsientoContable	int(11)	No			Clave primaria, autoincremental.

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
fecha_asiento	Date	Sí	NULL		Fecha en que se realizó el Asiento Contable.
Glosa	Text	Sí	NULL		Fecha en que se realizó el Asiento Contable.
Documento	varchar(15)	Sí	NULL		Documento relacionado con el asiento contable.
Observación	Text	Sí	NULL		Observación del Asiento Contable.
Estado	varchar(1)	Sí	NULL		Estado del Asiento Contable.
idPago	int(11)	Sí	NULL	pago -> idPago	Identificador del pago que origino el Asiento Contable.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifiko el registro.
dateUpdate	datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Sí N=No

Tabla 11. Estructura de la Tabla asientocontable
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idAsientoContable	2	A		
fk_asicont_pago_idPago	BTREE	No	No	idPago	2	A	YES	

Tabla 12. Indices de la Tabla asientocontable
Fuente: Elaborado por el Autor

TABLA: asientocontabledetalle

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idAsientoContableDetalle	int(11)	No			Clave primaria, autoincremental.
idAsientoContable	int(11)	Sí	NULL	asientocontable -> idAsientoContable	Identificador del Asiento Contable (padre).
nroCuenta	varchar(20)	Sí	NULL		Cuenta contable.

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
Importe	Doublé	Sí	NULL		Importe que se registrar para el Asiento Contable (hijo).
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 13. Estructura de la Tabla asientocontabledetalle

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idAsientoContableDetalle	2	A		
fk_asicontdet_asic ont_idAsientoCont able	BTREE	No	No	idAsientoContable	2	A	YES	

Tabla 14. Indices de la Tabla asientocontabledetalle

Fuente: Elaborado por el Autor

TABLA: beneficiario

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idBeneficiario	int(11)	No			Clave primaria, autoincremental.
Nombres	varchar(60)	Sí	NULL		Nombres del Beneficiario.
Apellidos	varchar(60)	Sí	NULL		Apellidos del Beneficiario.
Referencia	Longtext	Sí	NULL		Referencia y/o comentario que se quiere agregar sobre el Beneficiario.
Dni	varchar(8)	Sí	NULL		DNI (Documento Nacional de Identificación) del Beneficiario.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 15. Estructura de la Tabla beneficiario

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idBeneficiario	8	A		
iu_ben_dni	BTREE	Sí	No	Dni	8	A	YES	

Tabla 16. Índices de la Tabla beneficiario

Fuente: Elaborado por el Autor

TABLA: beneficiario_acreedor

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idAcreedor	int(11)	No		acreedor -> idAcreedor	Identificador del Acreedor.
idBeneficiario	int(11)	No		beneficiario -> idBeneficiario	Identificador del Beneficiario.
Relación	varchar(20)	Sí	NULL		Relación de parentesco entre el Acreedor y el Beneficiario.
userAdd	varchar(45)	Sí	NULL		Usuario que agregó el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 17. Estructura de la Tabla beneficiario_acreedor

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Si	No	idAcreedor	9	A		
				idBeneficiario	9	A		
fk_ben_idBen	BTREE	No	No	idBeneficiario	9	A		

Tabla 18. Índices de la Tabla beneficiario_acreedor
Fuente: Elaborado por el Autor

TABLA: deudaconcurasal

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idDeudaConcurasal	int(11)	No			Clave primaria, autoincremental.
Monto	Doublé	Sí	NULL		Monto total de la Deuda Concurasal.
Referencia	Longtext	Sí	NULL		Referencia y/o comentario que se desea agregar sobre la Deuda Concurasal.
idTipoDeuda	int(11)	Sí	NULL	tipodeuda -> idTipoDeuda	Tipo de Deuda Concurasal.
idAcreedor	int(11)	Sí	NULL	acreedor -> idAcreedor	Identificador del Acreedor.
idTipoDocumento	int(11)	Sí	NULL	tipodocumento -> idTipoDocumento	Identificador del Tipo de Documento que genero la Deuda Concurasal.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro.
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modificó el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 19. Estructura de la Tabla deudaconcurasal
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idDeudaConcurasa	3	A		

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
				1				
fk_deucon_acre_idAcreedor	BTREE	No	No	idAcreedor	3	A	YES	
fk_deucon_tipdeu_idTipoDeuda	BTREE	No	No	idTipoDeuda	3	A	YES	
fk_deucon_tipdoc_idTipoDocumento	BTREE	No	No	idTipoDocumento	3	A	YES	

Tabla 20. Indices de la Tabla deudaconcurasal
Fuente: Elaborado por el Autor

TABLA: pago

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idPago	int(11)	No			Clave primaria, autoincremental.
idProgramacionPago	int(11)	Si	NULL	programacionpago -> idProgramacionPago	Identificador de la Programación de Pago.
idAcreedor	int(11)	Si	NULL	acreedor -> idAcreedor	Identificador del Acreedor.
idBeneficiario	int(11)	Si	NULL	beneficiario -> idBeneficiario	Identificador del Beneficiario.
fecha_pago	Date	Si	NULL		Fecha en que se efectuó el pago.
Monto	Double	Si	NULL		Monto que se pagó.
Referencia	Text	No			Referencia y/o comentario.
userAdd	varchar(45)	Si	NULL		Usuario que agrego el registro
dateAdd	Datetime	Si	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Si	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Si	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Si	NULL		Indicador si el registro está activo. S=Si - Indicara si esta pagado N=No - Indicara si esta revertido
idTipoPago	int(11)	Si	NULL	tipopago -> idTipoPago	Identificador del Tipo de Pago en que se efectuaría el pago.

Tabla 21. Estructura de la Tabla pago
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idPago	2	A		
fk_pag_acre_idAcreedor	BTREE	No	No	idAcreedor	2	A	YES	
fk_pag_ben_idBeneficiario	BTREE	No	No	idBeneficiario	2	A	YES	
fk_pag_tippag_idTipoPago	BTREE	No	No	idTipoPago	2	A	YES	
fk_pag_propag_idProgramacionPago	BTREE	No	No	idProgramacionPago	2	A	YES	

Tabla 22. Índices de la Tabla pago
Fuente: Elaborado por el Autor

TABLA: parametros generales

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
Id	int(11)	No			Identificador, autoincremental.
Igv	double(15,3)	Sí	NULL		Impuesto General a las Ventas
Uit	double(15,3)	Sí	NULL		Unidad Impositiva Tributaria
monto_minimo_pagar	double(15,3)	Sí	NULL		Monto mínimo que se pagará por Acreencia
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 23. Estructura de la Tabla parametros_generales
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	Id	1	A		

Tabla 24. Índices de la Tabla parametros_generales
Fuente: Elaborado por el Autor

TABLA: programacionpago

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idProgramacionPago	int(11)	No			Clave primaria, autoincremental.
idAcreencia	int(11)	Sí	NULL	acreencia -> idAcreencia	Identificador de la Acreencia.
nro_cuotas	int(11)	Sí	NULL		Numero de cuotas que se programan para pagar.
Mesanio	Date	Sí	NULL		Semestre compuesto por mes y año que se programa el pago.
Importe	Double	Sí	NULL		Importe que se paga.
Estado	varchar(1)	Sí	NULL		Estado de la Programación de Pago: 1=Programado, 2=Reprogramado.
Referencia	Longtext	Sí	NULL		Referencia y/o comentario que se desea agregar en la Programación de Pago.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 25. Estructura de la Tabla programacionpago
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idProgramacionPago	10	A		

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
fk_propag_acre	BTREE	No	No	idAcreencia	2	A	YES	

Tabla 26. Índices de la Tabla programacionpago
Fuente: Elaborado por el Autor

TABLA: proveedor

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
Ruc	varchar(11)	No			Clave primaria, (Registro Único de Contribuyente).
razon_social	varchar(80)	Sí	NULL		Razón social del Acreedor.
nombre_comercial	varchar(80)	Sí	NULL		Nombre comercial del Acreedor (Opcional)
Estado	varchar(1)	Sí	NULL		Estado del Proveedor.
Celular	varchar(11)	Sí	NULL		Celular del Proveedor.
Teléfono	varchar(11)	Sí	NULL		Teléfono del Proveedor.
Dirección	Longtext	Sí	NULL		Dirección del Proveedor.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	No			Indicador si el registro está activo. S=Sí N=No
idAcreedor	int(11)	Sí	NULL	acreedor -> idAcreedor	Identificador del Acreedor

Tabla 27. Estructura de la Tabla proveedor
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	ruc	3	A		
fk_proveed_acreed_idAcreedor	BTREE	No	No	idAcreedor	3	A	YES	

Tabla 28. Índices de la Tabla proveedor
Fuente: Elaborado por el Autor

TABLA: tipoacreencia

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idTipoAcreencias	int(11)	No			Clave primaria, autoincremental.
Descripción	Longtext	Sí	NULL		Descripcion y/o nombre del Tipo de Deuda Concursal.
idTipoDeuda	int(11)	Sí	NULL	tipodeuda -> idTipoDeuda	Identificador del Tipo de Deuda Concursal.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifiko el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Sí N=No

Tabla 29. Estructura de la Tabla tipoacreencia

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idTipoAcreencias	9	A		
fk_tipacre_tipden_idTipoDeuda	BTREE	No	No	idTipoDeuda	4	A	YES	

Tabla 30. Indices de la Tabla tipoacreencia

Fuente: Elaborado por el Autor

TABLA: tipodeuda

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idTipoDeuda	int(11)	No			Clave primaria, autoincremental.
Descripción	Longtext	Sí	NULL		Descripcion y/o nombre del Tipo de Deuda.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro.
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifiko el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 31. Estructura de la Tabla tipodeuda
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idTipoDeuda	2	A		

Tabla 32. Indices de la Tabla tipodeuda
Fuente: Elaborado por el Autor

TABLA: tipodocumento

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idTipoDocumento	int(11)	No			Clave primaria, autoincremental.
Descripción	Longtext	Sí	NULL		Descripción y/o nombre del Tipo de Documento que genero la Deuda Concursal.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifíco el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 33. Estructura de la Tabla tipodocumento
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idTipoDocumento	4	A		

Tabla 34. Indices de la Tabla tipodocumento
Fuente: Elaborado por el Autor

TABLA: tipoestado

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idTipoEstado	int(11)	No			Clave primaria, autoincremental.
descripcion	Longtext	Sí	NULL		Descripción y/o nombre del Tipo de Estado de una Acreencia.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 35. Estructura de la Tabla tipoestado

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idTipoEstado	4	A		

Tabla 36. Indices de la Tabla tipoestado

Fuente: Elaborado por el Autor

TABLA: tipopago

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
idTipoPago	int(11)	No			Clave primaria, autoincremental.
Descripción	Longtext	Sí	NULL		Descripción y/o nombre del Tipo de Pago.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro.
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 37. Estructura de la Tabla tipopago

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	idTipoPago	4	A		

Tabla 38. Indices de la Tabla tipopago
Fuente: Elaborado por el Autor

TABLA: trabajador

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
Dni	varchar(8)	No			Clave primaria, (Documento Nacional de Identificación).
Nombres	varchar(60)	Sí	NULL		Nombres del Acreedor.
Apellidos	varchar(60)	Sí	NULL		Apellidos del Acreedor.
Estado	varchar(1)	Sí	NULL		Estado de un Trabajador.
Celular	varchar(11)	Sí	NULL		Celular del Trabajador.
Teléfono	varchar(11)	Sí	NULL		Telefono del Trabajador.
Dirección	Longtext	Sí	NULL		Direccion del Proveedor.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Sí	NULL		Indicador si el registro está activo. S=Si N=No
idAcreedor	int(11)	Sí	NULL	acreedor -> idAcreedor	Identificador de Acreedor.

Tabla 39. Estructura de la Tabla trabajador
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	Dni	5	A		

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
fk_trab_acree_idAcreeedor	BTREE	No	No	idAcreeedor	5	A	YES	

Tabla 40. Indices de la Tabla trabajador
Fuente: Elaborado por el Autor

TABLA: users

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
Iduser	int(11)	No			Clave primaria.
Nombres	varchar(50)	No			Nombres del usuario que ingresara al sistema.
Apellidos	varchar(50)	No			Apellidos del usuario que ingresara al sistema.
Username	varchar(45)	No			Nombre de usuario que ingresara al sistema.
Password	varchar(45)	No			Contraseña del usuario que ingresara al sistema.
Enabled	bit(1)	Sí	NULL		Campo que identifica si está activo (true) o no (false) el usuario.
userAdd	varchar(45)	Sí	NULL		Usuario que agrego el registro
dateAdd	Datetime	Sí	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Sí	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Sí	NULL		Fecha en que se modificó el registro.

Tabla 41. Estructura de la Tabla users
Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Sí	No	iduser	5	A		
users_username_key	BTREE	Sí	No	username	5	A		

Tabla 42. Indices de la Tabla users
Fuente: Elaborado por el Autor

TABLA: user_roles

CAMPO	TIPO	NULO	PREDETERMINADO	ENLACES A	COMENTARIOS
user_role_id	int(11)	No			Clave primaria.
Username	varchar(45)	No		users -> username	Nombres de usuario que ingresara al sistema.
Role	varchar(45)	No			Role del usuario que ingresara al sistema.
userAdd	varchar(45)	Si	NULL		Usuario que agrego el registro
dateAdd	Datetime	Si	NULL		Fecha en que se agregó el registro.
userUpdate	varchar(45)	Si	NULL		Usuario que modifico el registro.
dateUpdate	Datetime	Si	NULL		Fecha en que se modificó el registro.
indActivo	varchar(1)	Si	NULL		Indicador si el registro está activo. S=Si N=No

Tabla 43. Estructura de la Tabla user_roles

Fuente: Elaborado por el Autor

ÍNDICES:

NOMBRE DE LA CLAVE	TIPO	ÚNICO	EMPACADO	CAMPO	CARDINALIDAD	COTEJAMIENTO	NULO	COMENTARIO
PRIMARY	BTREE	Si	No	user_role_id	5	A		
user_roles_username_fkey	BTREE	No	No	username	5	A		

Tabla 44. Índices de la Tabla user_roles

Fuente: Elaborado por el Autor

4.2.2. MODELAO DE DATOS

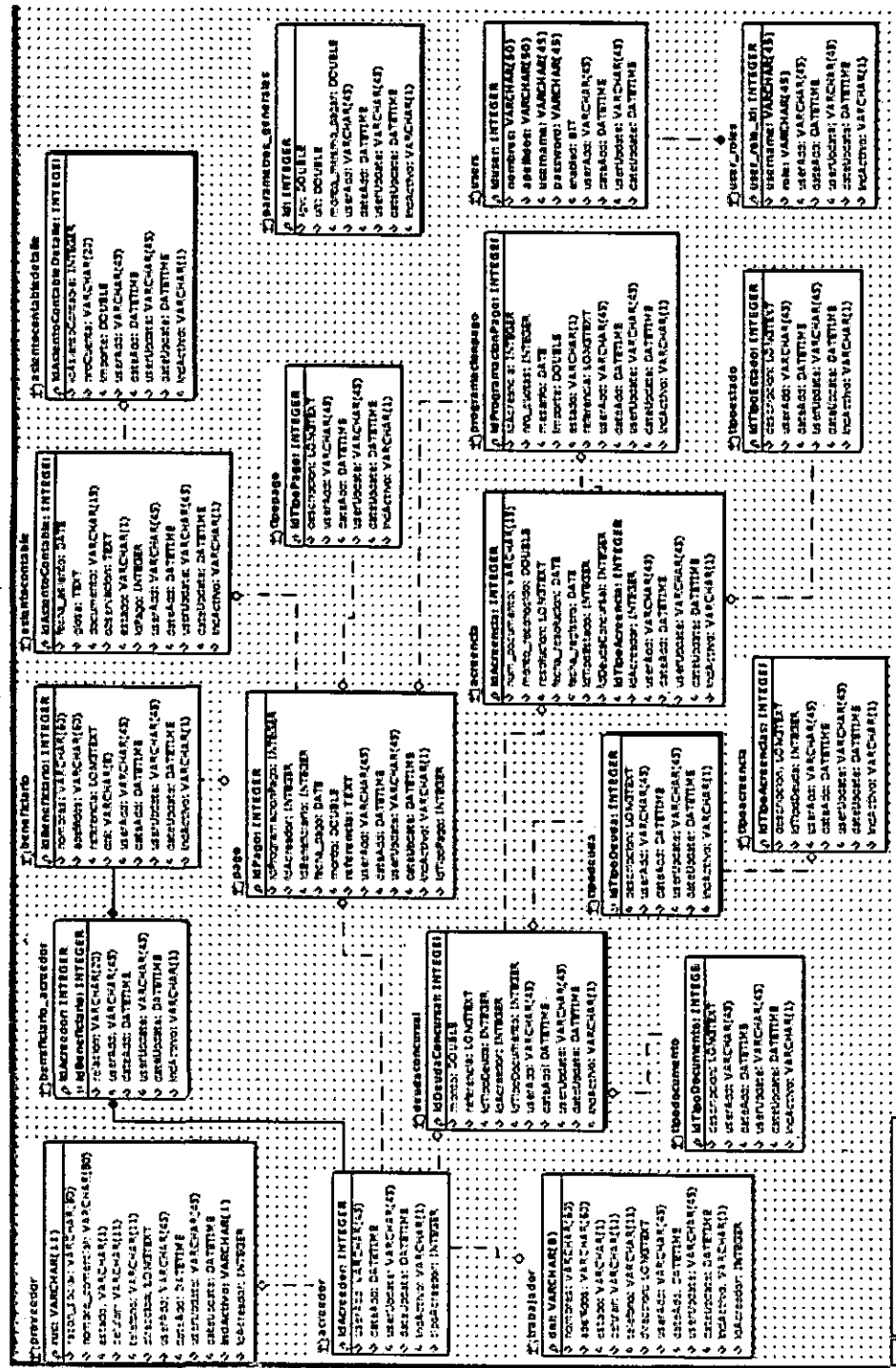


Figura 4. Diseño de la Base de Datos
Fuente: Elaborado por el autor

4.3. DISEÑO DE LA INTERFAZ GRÁFICA

SICA utiliza la tecnología de **JavaServer Pages (JSP)** porque permite crear páginas web dinámicas independientemente de la plataforma, además separa la interfaz de usuario de la generación de contenidos, permitiendo cambiar el diseño total de las páginas sin alterar el contenido dinámico subyacente. **(ANEXO 2)**

Para la creación de las interfaces se tomó como referencia el concepto de **USABILIDAD** de la **ISO / IEC 2500** o estándar **SQuaRE (Requisitos y evaluación de la calidad de producto software)**, la cual debe cumplir con lo siguiente:

- **Capacidad de aprendizaje.** Capacidad de SICA permita al usuario aprender su aplicación.
- **Capacidad para ser usado.** Capacidad de SICA permita al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad SICA para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad de SICA permita que sea utilizado por usuarios con determinadas características.

4.3.1. AUTENTICACIÓN.

Para poder acceder al SICA previamente debe autenticarse ingresando su usuario y contraseña válidos, por lo tanto nuestra interfaz solicita esos campos:

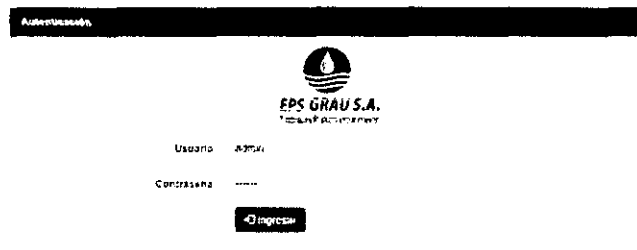


Figura 5. Interfaz del Logueo
Fuente: Elaborado por el Autor

4.3.2. MENU PRINCIPAL.

Luego de ingresar al SICA y dependiendo del rol que posea el usuario se le presentará su respectivo menú de opciones y para ambos casos contara con la

opción Cerrar Sesión para salir del sistema el cual nos llevar a la interfaz de autenticación.

4.2.2.1. ROL ADMINISTRADOR.

Para un usuario de Rol Administrador se le presentará los menú de Parametría, Acciones especiales, Registrar y Reporte; este posee el control sobre SICA por tener más accesos y permisos.

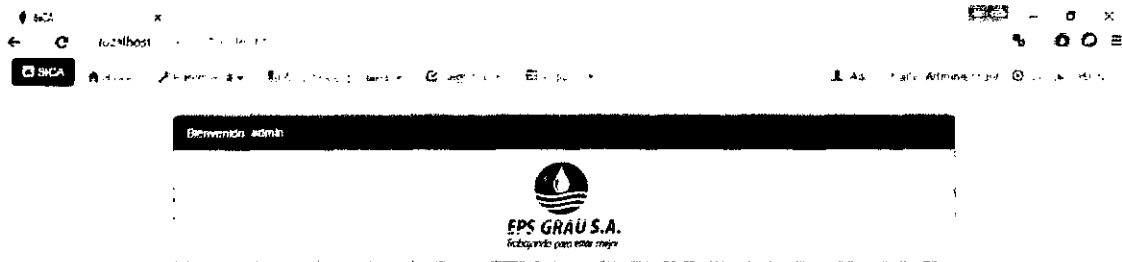


Figura 6. Menú para el Usuario Tipo Administrador
Fuente: Elaborado por el Autor

4.2.2.2. ROL USUARIO ESTANDAR.

Para un usuario de Rol estándar solo se le presentara el menú de Reporte, porque este solo se le permite realizar consultas.



Figura 7. Menú para el Usuario Tipo Normal
Fuente: Elaborado por el Autor

4.3.3. MENÚ: PARAMETRÍA.

Este menú se creó con la finalidad que en éste se realicen el mantenimiento de lo que corresponda a la parametría del SICA. Lo submenús que posee se muestra en la siguiente imagen:

✎ Parametría ▾

Parámetros Generales

Usuario

Tipo Documento

Tipo Pago

Tipo Deuda

Tipo Acreencia

Tipo Estado de Acreencia

Figura 8. Menú de Parametría
Fuente: Elaborado por el Autor

4.2.3.1. PARÁMETRO GENERALES

Esta interfaz es para realizar el mantenimiento de los parámetros generales, los campos que se muestran con (*) están considerados como obligatorios y de estar en blanco y querer hacer algún cambio el SICA no lo permitirá y mostrará un mensaje de alerta¹, el campo ID solo de lectura por lo tanto no se permite agregar valor alguno².

Parámetros Generales

ID.

IGV (*)

UIT (*)

Monto Máximo Pagar (*)

Listar

(*) Campos obligatorios.

Figura 9. Interfaz de Mantenimiento de Parámetros Generales
Fuente: Elaborado por el Autor

Al hacer click en el botón **listar**³ se nos presentará una tabla mostrándonos los valores respectivos y además en la última columna en cada fila tendremos un botón editar en el cual al hacer click en él nos permitirá modificar los valores del elemento de la fila respectiva:

¹ La misma funcionalidad que explica será igual en todo las interfaces donde los campos tengan (*).

² Todos los campos ID. Será solo de lectura.

³ La función de mostrar una tabla será igual para todos interfaces que tengan un botón Listar.

Fuente: Elaborado por el Autor

Figura 11. Interfaz de Parámetros Generales luego de realizar la acción Editar
Fuente: Elaborado por el Autor

⁴ La función de regresarnos a un estado inicial con un formulario desactivado y con campos en blanco será la misma para todas las interfaces que cuenten con el botón Cancelar.

4.2.3.2. USUARIO.

Esta interfaz es para realizar el mantenimiento de los usuarios permitiendo realizar el **CRUD** (Crear, Leer, Actualizar y Borrar teniendo en cuenta que se realiza una eliminación lógica mediante el **checkbox Indicador Activo**), en primer lugar el formula aparece con los campos desactivados solo con los botones **Nuevo** y **Lista** activos.

Registrar Usuario

ID. Usuario

Nombres (*)

Apellidos (*)

Tipo Usuario (*) Seleccione tipo

Usuario (*)

Contraseña (*)

Indicador Activo

Nuevo Listar

(*) Campos obligatorios

Figura 12. Interfaz de Mantenimiento de un Usuario
Fuente: Elaborado por el Autor

Al hacer click en el **botón Nuevo**⁵ tendremos los campos activos para ingresar los valores correspondientes y nos mostrará lo siguiente:

Registrar Usuario

ID. Usuario

Nombres (*)

Apellidos (*)

Tipo Usuario (*) Seleccione tipo

Usuario (*)

Contraseña (*)

Indicador Activo

Registrar Cancelar Listar

(*) Campos obligatorios.

Figura 13. Interfaz de Usuario luego de hacer click en Nuevo.
Fuente: Elaborado por el Autor

⁵ La función de mostrarnos un formulario activo para ingresar valores será la misma para todas las interfaces que cuente con el botón Nuevo.

En el campo Tipo Usuario permite escoger entre Administrador o Estándar que será el tipo de rol que tendrá el usuario a crear y/o modificar.

4.2.3.3. TIPO DOCUMENTO.

Esta interfaz es para realizar el mantenimiento de un Tipo Documento permitiendo realizar el **CRUD (Crear, Leer, Actualizar y Borrar)** teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox Indicador Activo)⁶.

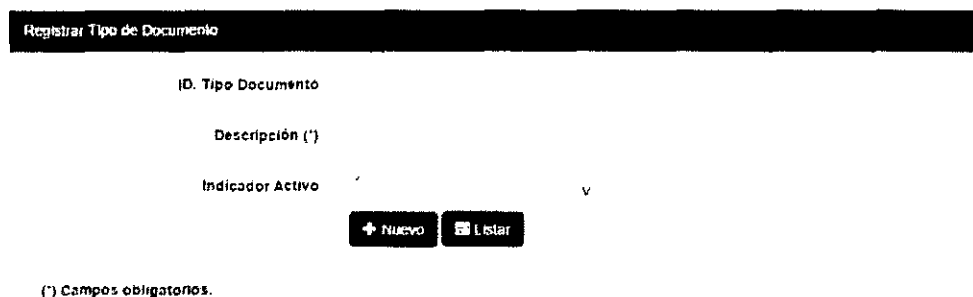


Figura 14. Interfaz de Mantenimiento de un Tipo de Documento
Fuente: Elaborado por el Autor

4.2.3.4. TIPO PAGO.

Esta interfaz es para realizar el mantenimiento de un Tipo Pago permitiendo realizar el **CRUD (Crear, Leer, Actualizar y Borrar)** teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox Indicador Activo)⁷.

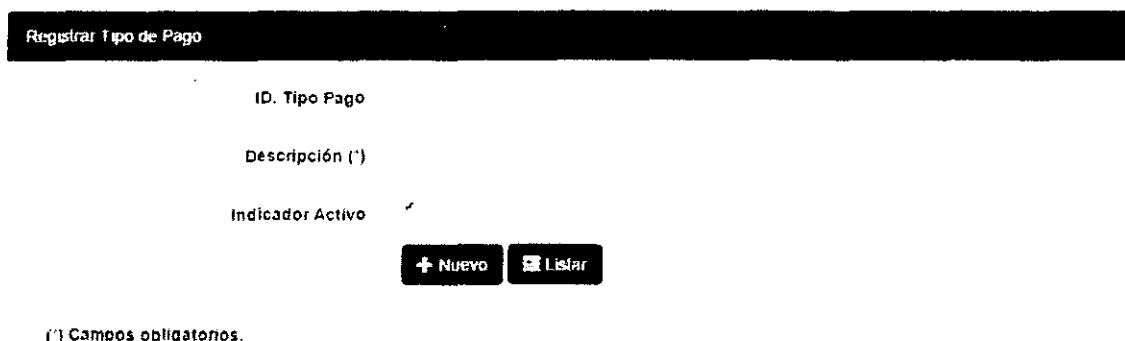


Figura 15. Interfaz de Mantenimiento de un Tipo Pago
Fuente: Elaborado por el Autor

⁶ Los valores que se registren en esta interfaz alimentará los campos select Tipo Documento donde se le requiera.

⁷ Los valores que se registren en esta interfaz alimentará los campos select Tipo Pago donde se le requiera.

4.2.3.5. TIPO DEUDA.

Esta interfaz es para realizar el mantenimiento de un Tipo Deuda permitiendo realizar el **CRUD** (Crear, Leer, Actualizar y Borrar teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox Indicador Activo)⁸.

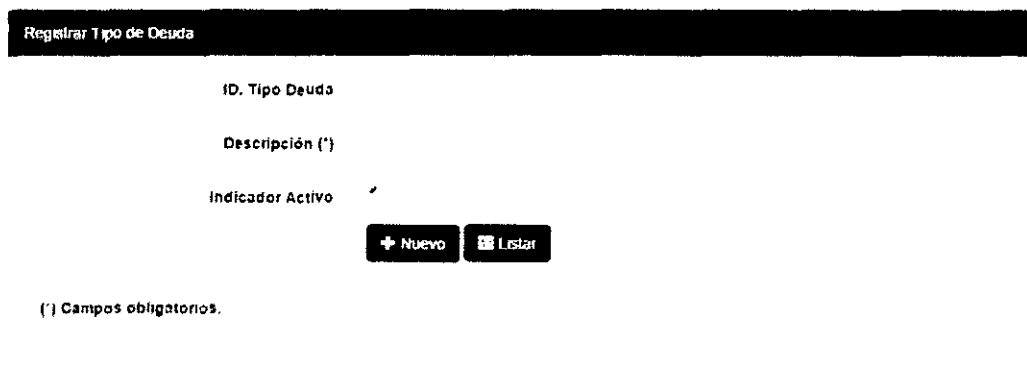


Figura 16. Interfaz de Mantenimiento de un Tipo Deuda
Fuente: Elaborado por el Autor

Para el caso de **SICA** y producto del proceso concursal de la EPS **GRAU** se tendrá **LABORAL** y **COMERCIAL**.

4.2.3.6. TIPO ACREENCIA.

Esta interfaz es para realizar el mantenimiento de un Tipo Acreencia permitiendo realizar el **CRUD** (Crear, Leer, Actualizar y Borrar teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox Indicador Activo)⁹.

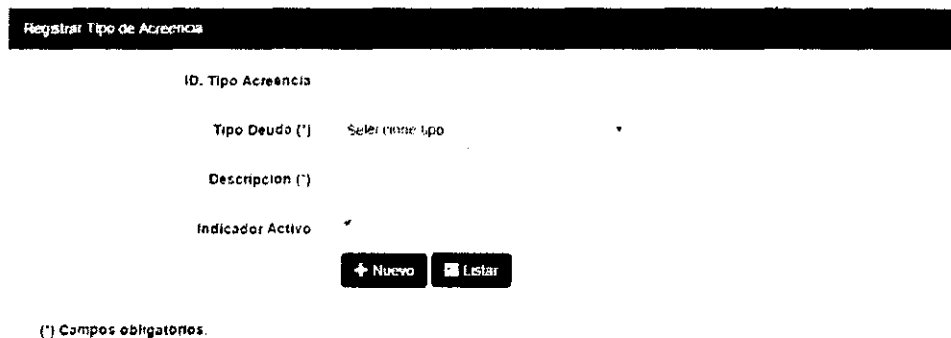


Figura 17. Interfaz de Mantenimiento de un Tipo Acreencia
Fuente: Elaborado por el Autor

⁸ Los valores que se registren en esta interfaz alimentará los campos select Tipo Deuda donde se le requiera.

⁹ Los valores que se registren en esta interfaz alimentará los campos select Tipo Acreencia donde se le requiera.

Para SICA y producto del proceso concursal de la EPS GRAU en el caso del tipo deuda **LABORAL** tendrá los tipo de acreencias: CTS, beneficios laborales, procesos laborales, colaterales y procesos de viudez; y para el tipo deuda **COMERCIAL** tendrá los tipo de acreencias: comerciales, procesos civiles, tributarios y contingentes.

4.2.3.7. TIPO ESTADO DE ACREENCIA.

Esta interfaz es para realizar el mantenimiento de un Tipo Estado de Acreencia permitiendo realizar el **CRUD (Crear, Leer, Actualizar y Borrar)** teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox **Indicador Activo**)¹⁰.

Producto del proceso concursal de la EPS GRAU SICA tendrá registrado **RECONOCIDO y NO RECONOCIDO**

Figura 18. Interfaz de Mantenimiento de un Tipo Estado de Acreencia
Fuente: Elaborado por el Autor

4.3.4. MENÚ: ACCIONES ESPECIALES

Este menú se creó con la finalidad de que contenga lo relacionado a acciones especiales donde no se realiza un CRUD, se cuenta con los submenús que se muestra en la siguiente imagen:

¹⁰ Los valores que se registren en esta interfaz alimentará los campos select Estado de Acreencia donde se le requiera.

Acciones Especiales ▾

Camblar Tipo Acreencia

Extornar Pagos

Figura 19. Menú de Acciones Especiales
Fuente: Elaborado por el Autor

4.2.4.1. CAMBIAR TIPO ACREENCIA

Esta interfaz es para realizar el cambio de tipo de una Acreencia en el cual en primer lugar se debe ingresar un número de acreencia donde se mostrará sus datos correspondiente y de no existir y/o estar inactivo aparecerá una alerta.

Figura 20. Interfaz para el Cambio Tipo Acreencia
Fuente: Elaborado por el Autor

4.2.4.2. EXTORNAR PAGOS

En esta interfaz se realizan los extornos de algún pago que se haya realizado por error. Luego de ingresar los datos respectivos nos aparecerá un alerta indicándonos si el proceso se realizó o no.

Figura 21. Interfaz para Extorna Pagos Realizados
Fuente: Elaborado por el Autor

4.3.5. MENU: REGISTRAR.

Este menú se creó con la finalidad que contenga los procesos que son cruciales para SICA y en el cual gira toda la lógica del negocio, contiene los submenús que se presentan en la siguiente imagen:

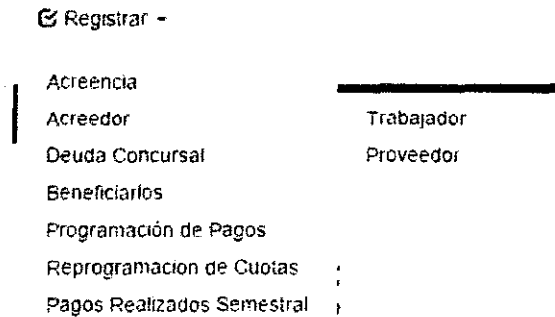


Figura 22. Menú de Registrar
Fuente: Elaborado por el Autor

4.2.5.1. ACREENCIA.

Esta interfaz es para realizar el mantenimiento de una Acreencia permitiendo realizar el **CRUD (Crear, Leer, Actualizar y Borrar)** teniendo en cuenta que se realiza una eliminación lógica mediante el **checkbox Indicador Activo**). En primer lugar debemos ingresar un Nro. De Deuda Concursal y hacer click en buscar y nos mostrará la información correspondiente para continuar con el proceso caso contrario nos mostrará un mensaje que no se encontraron resultados.

Nueva Acreencia

Nro. Acreencia

Nro. Deuda Concursal (*)

Acreedor

Tipo Acreedor

Tipo Deuda

Tipo Acreencia Seleccione Tipo

Nro. Documento (*)

Monto Reconocido (*)

Resolución (*)

Fecha Resolución (*)

Fecha Registro (*)

Indicador Activo

(*) Campos obligatorios.

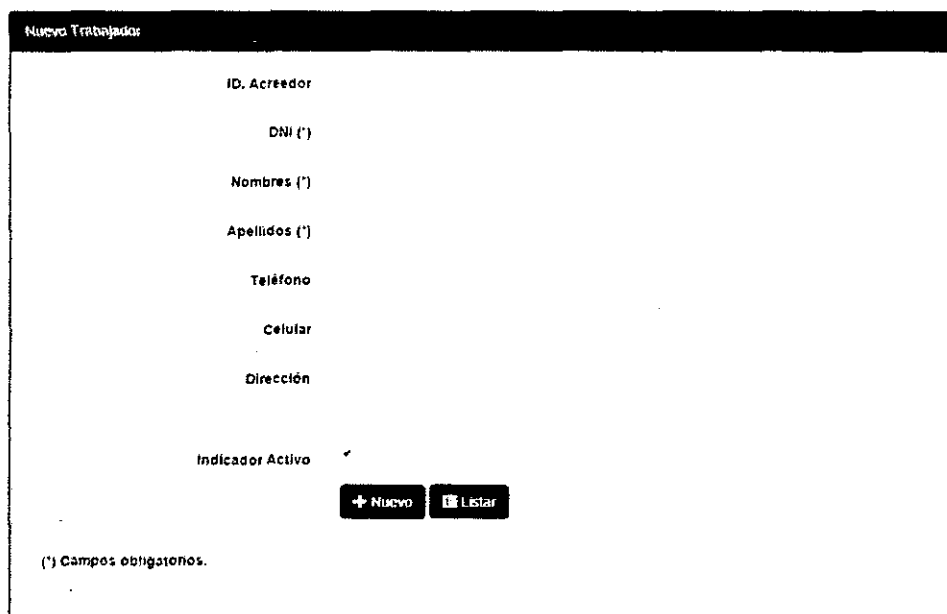
Figura 23. Interfaz de Mantenimiento de una Acreencia
Fuente: Elaborado por el Autor

4.2.5.2. *ACREEDOR.*

Este submenú se creó con la finalidad que contenga los tipos de acreedores que generó el proceso concursal que son trabajadores y proveedores y que deben ser administrados por SICA.

4.2.4.2.1. *TRABAJADOR.*

Esta interfaz es para realizar el mantenimiento de un Trabajador permitiendo realizar el **CRUD (Crear, Leer, Actualizar y Borrar)** teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox **Indicador Activo**).



Nuevo Trabajador

ID, Acreedor

DNI (*)

Nombres (*)

Apellidos (*)

Teléfono

Celular

Dirección

Indicador Activo ☐

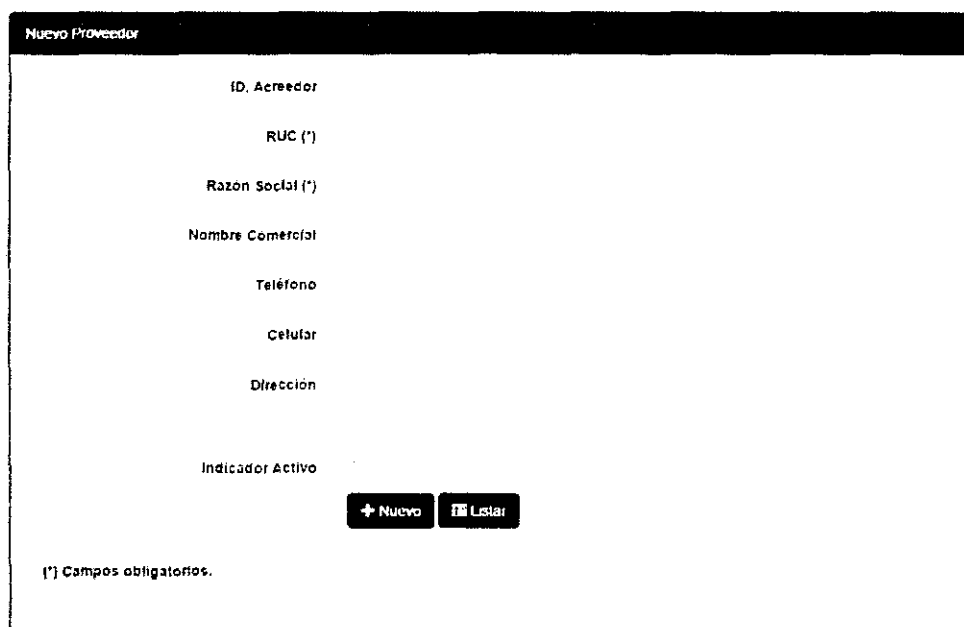
+ Nuevo Listar

(*) Campos obligatorios.

Figura 24. Interfaz de Mantenimiento de un Trabajador
Fuente: Elaborado por el Autor

4.2.4.2.2. *PROVEEDOR*

Esta interfaz es para realizar el mantenimiento de un Proveedor permitiendo realizar el **CRUD (Crear, Leer, Actualizar y Borrar)** teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox **Indicador Activo**).



Nuevo Proveedor

ID, Acreedor

RUC (*)

Razón Social (*)

Nombre Comercial

Teléfono

Celular

Dirección

Indicador Activo ☐

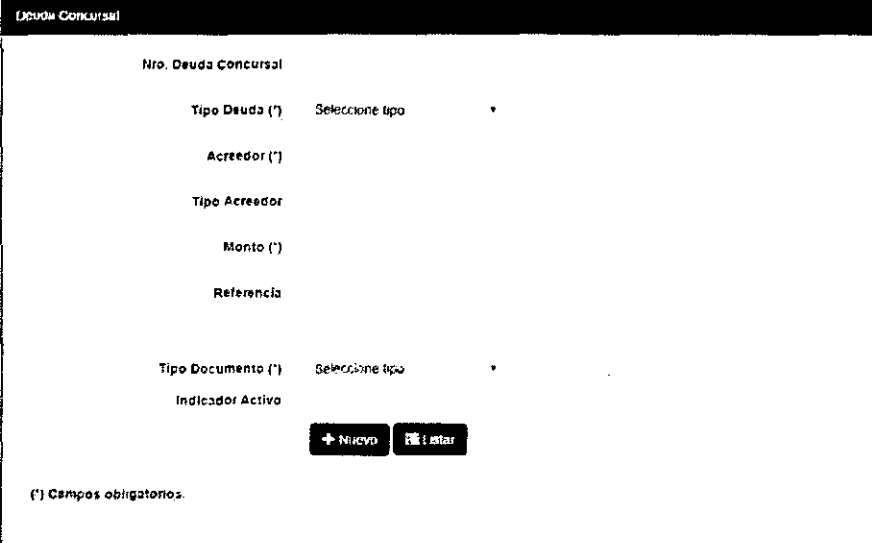
+ Nuevo Listar

(*) Campos obligatorios.

Figura 25. Interfaz de Mantenimiento de un Proveedor
Fuente: Elaborado por el Autor

4.2.5.3. DEUDA CONCURSAL.

Esta interfaz es para realizar el mantenimiento de una Deuda Concursal permitiendo realizar el **CRUD** (Crear, Leer, Actualizar y Borrar teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox Indicador Activo).



Deuda Concursal

Nro. Deuda Concursal

Tipo Deuda (*) Seleccione tipo *

Acreedor (*)

Tipo Acreedor

Monto (*)

Referencia

Tipo Documento (*) Seleccione tipo *

Indicador Activo

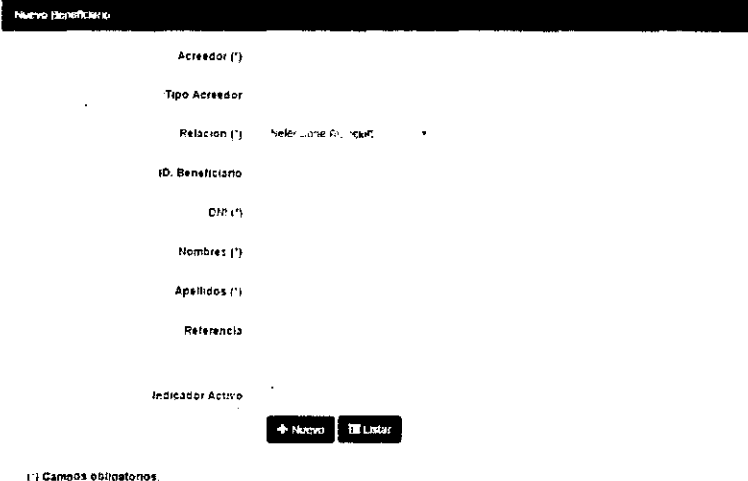
+ Nuevo Eliminar

(*) Campos obligatorios.

Figura 26. Interfaz de Mantenimiento de una Deuda Concursal
Fuente: Elaborado por el Autor

4.2.5.4. BENEFICIARIOS.

Esta interfaz es para realizar el mantenimiento de un Beneficiario permitiendo realizar el **CRUD** (Crear, Leer, Actualizar y Borrar teniendo en cuenta que se realiza una eliminación lógica mediante el checkbox Indicador Activo).



Nuevo Beneficiario

Acreedor (*)

Tipo Acreedor

Relacion (*) Seleccione relacion *

ID Beneficiario

DNI (*)

Nombres (*)

Apellidos (*)

Referencia

Indicador Activo

+ Nuevo Eliminar

(*) Campos obligatorios.

Figura 27. Interfaz de Mantenimiento de un Beneficiario
Fuente: Elaborado por el Autor

4.2.5.5. PROGRAMACIÓN DE PAGOS.

Esta interfaz es para realizar el proceso de la Programación de Pagos. Previamente se debe realizar la búsqueda de una Acreencia de existir presentará los valores correspondientes caso contrario nos enviará al estado inicial del formulario.

Luego de ingresar todos los campos requeridos y hacer click en programar nos aparecerá un alerta informándonos si el proceso se ejecutó correctamente o si es que ya se ha realizado una programación para esa acreencia.

Se programan (02) pagos al año en el mes de Abril y Septiembre.

Programación de Pagos

Nro. Acreencia (*)

Acreedor

Tipo Acreedor

Tipo Deuda

Tipo Acreencia

Estado Acreencia

Importe (*)

Nro. Cuotas (*)

Desde (*)

Referencia

+ Nuevo

(*) Campos obligatorios.

Figura 28. Interfaz para el Proceso de Programación de Pagos
Fuente: Elaborado por el Autor

4.2.5.6. REPROGRAMACIÓN DE PAGOS.

Esta interfaz es para realizar el proceso de la Reprogramación de Pagos. Previamente se debe realizar la búsqueda de una Acreencia de existir presentará los valores correspondientes caso contrario nos presentará un alerta debido que solo se debe buscar acreencias en estado reconocidas y que existan.

Luego de ingresar todos los campos requeridos y hacer click en reprogramar nos aparecerá un alerta informándonos si el proceso se ejecutó correctamente o si es que ya se ha realizado una reprogramación para esa acreencia.

Se programan (02) pagos al año en el mes de Abril y Septiembre.

Figura 29. Interfaz para el Proceso de Reprogramación de Pagos
Fuente: Elaborado por el Autor

4.2.5.7. **PAGOS REALIZADOS SEMESTRAL.**

Esta interfaz es para realizar el proceso de Pagos. Previamente debemos buscar una programación con el Nro. De Acreencia y Mes / Año.

Luego de ingresar la cuota se nos presentará los valores correspondientes caso contrario nos mostrará una alerta informándonos si ya se realizó el pago de esa cuota, si no se encontraron resultados o si se encuentra reprogramada.

También para poder realizar un pago debemos escoger a un beneficiario para el cual se nos permite realizar la búsqueda por Nro.

DNI y/o ingresando sus apellidos o nombres, presentándonos una ventana emergente.

Pagos Realizados Semestral

Nro. Acreencia Q. Buscar

Mes / Año

Nro. Programación Pago (*)

Acreedor (*)

Tipo Acreedor

Beneficiario Buscar Pto.

Relación

Monto (*)

Fecha Pago (*)

Tipo Pago (*) Seleccione tipo

Referencia

➕ Nuevo

(*) Campos obligatorios

Figura 30. Interfaz para el Proceso de Pagos Realizados Semestralmente
Fuente: Elaborado por el Autor

Búsqueda de Beneficiarios

DNI: Buscar

Nº	NOMBRE	RELACION	PASAR
1	CARLOS NOLE	Otro	<input checked="" type="checkbox"/>

Close

Figura 31. Interfaz para la búsqueda de beneficiario por Nro. DNI
Fuente: Elaborado por el Autor

Búsqueda de Beneficiarios

Nombres y/o Apellidos:		car	Buscar
Nº	NOMBRE	RELACION	PASAR
1	CARMEN CYNTHIA ELIZABETH RAMOS ARCA	Esposo (a)	<input checked="" type="checkbox"/>
2	CARLOS NOLE	Otro	<input checked="" type="checkbox"/>
			Close

Figura 32. Interfaz para realizar la búsqueda de un beneficiario por Nombre y/o Apellidos
Fuente: Elaborado por el Autor

4.3.6. MENU REPORTE.

Este menú tiene como finalidad brindar los reportes necesarios para que a los usuarios que utilicen SICA mediante la información brindada le ayude a una mejor toma de decisiones.

Solo se mostrará el diseño de los reportes de los procesos más importantes del proceso concursal de la EPS GRAU. En todos los reportes que genere SICA tendrá el código del usuario que genera el reporte, la fecha y hora en que se genera el reporte y el número de páginas.

El diseño para la salida en formato XLS es el mismo que el de formato PDF. SICA se podrá generar reportes en formato PDF y XLS.

Reporte -

Deuda Concursal

Pagos Realizados Semestral

Pagos Extornados

Programación Pagos

Acreencias

Acreedor

Beneficiario

Trabajador

Proveedor

Figura 33. Menú de Reportes
Fuente: Elaborado por el Autor

4.2.5.1. DEUDA CONCURSAL.

En esta interfaz se puede generar el Reporte de Deuda Concursal y tendrá como criterios de búsqueda: tipo deuda y acreedor.

Reporte Deuda Concursal

Criterio (*) Deuda ▼
 Tipo Deuda Seleccione tipo ▼

(*) Campos obligatorios.

Figura 34. Interfaz para generar Reporte de Deuda Concursal
Fuente: Elaborado por el Autor

REPORTE DEUDA CONCURSAL			
ACREDITAR	TIPO DOCUMENTO	REFERENCIA	MONTOS DEUDAS
ACREDITAR	REFERENCIA	REFERENCIA	
	CONCURSAL		
1	PLANILLA		15000
ACREDITAR	CASA MARCELA GARCIA Y OROZCO S.A.S.		
	TR. DEUDA		
11	FACTURA		10000

15/01/2016 6:44 PM

Página 1 de 1

Figura 35. Reporte de Deuda Concursal en formato PDF
Fuente: Elaborado por el Autor

USUARIO admin

SICA vs 10

ID	PAGO	FECHA PAGO	MONTO	REFERENCIA	TIPO PAGO	ID ACREEDOR	ACREEDOR	TIPO ACREEDOR	ID BENEF	BENEFICIARIO	NRO. APROVECHA	NRO. CUOTA	MES/AÑO	ESTADO
27		08/01/2016	1428.5714		CHEQUE	1	CARLOS NOLE	TRABAJADOR	19	CARLOS NOLE	1	2	09/2016	PROGRAMACION
28		08/01/2016	1428.5714		GIRO	1	CARLOS NOLE	TRABAJADOR	11	SOFIA MACHACA	1	1	03/2016	PROGRAMACION

Figura 38. Reporte de Pagos Realizados Semestralmente en formato PDF
Fuente: Elaborado por el Autor

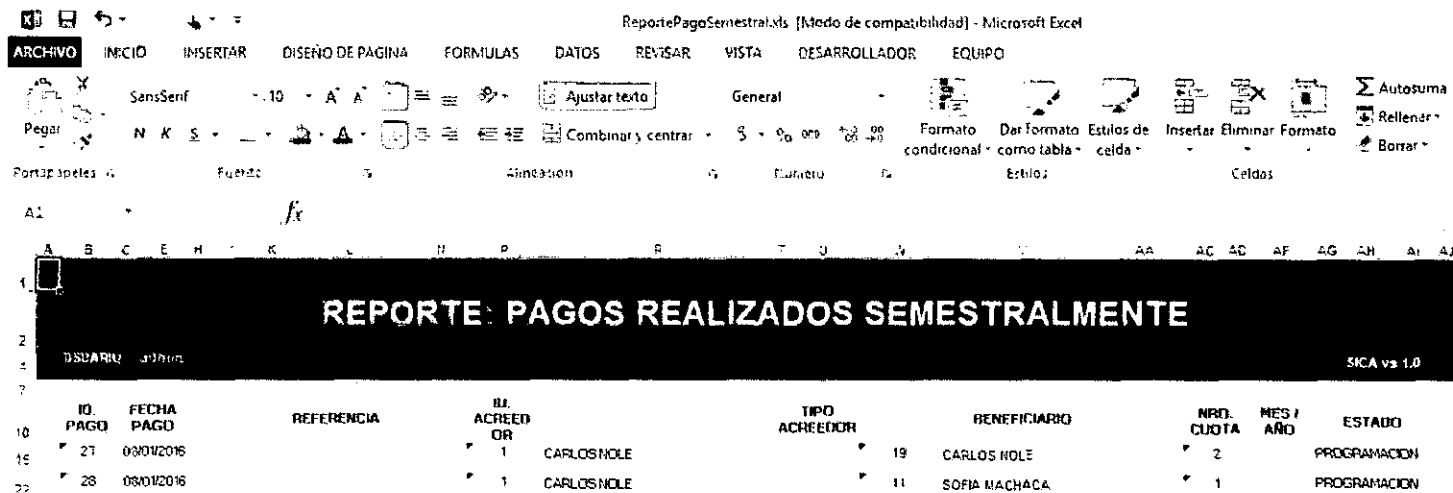


Figura 39. Reporte de Pagos Realizados Semestralmente en formato XLS
Fuente: Elaborado por el Autor

4.2.5.3. PAGOS EXTORNADOS

En esta interfaz se puede generar el Reporte de Pagos Extornados y tendrá como criterios de búsqueda: Nro. Acreencia y rango de fechas.

Reporte: Pagos Extornados

Criterio (*) Rango Fechas

Desde 12-01-2016

Hasta 12-01-2016

Exportar a Excel Exportar a PDF

(*) Campos obligatorios.

Figura 40. Interfaz para generar el Reporte de Pagos Extornados
Fuente: Elaborado por el Autor

4.2.5.4. PROGRAMACIÓN DE PAGOS

En esta interfaz se puede generar el Reporte de Programación de Pagos y tendrá como criterios de búsqueda: Mes / Año, Nro. Acreencia y estado (Programado, Reprogramado o Todos).

Reporte: Programación de Pagos

Criterio (*) Rango Fechas

Desde 12-01-2016

Exportar a Excel Exportar a PDF

(*) Campos obligatorios.

Figura 41. Interfaz para generar el Reporte de Programación de Pagos
Fuente: Elaborado por el Autor

REPORTE: PROGRAMACION DE PAGOS

USUARIO admin

SICA vs 1.0

ID PROG PAGO	NRO CUOTA	MES / AÑO	IMPORTE	ACTIVO	ESTADO	REFERENCIA	ID ACREEDOR	ACREEDOR	NRO ACREENCIA	TIPO DEUDA	TIPO ACREENCIA	ESTADO ACREENCIA
133	1	03/2016	1428.5714	SI	PROGRAMADO		1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
134	2	09/2016	1428.5714	SI	PROGRAMADO		1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
135	3	03/2017	1428.5714	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
136	4	09/2017	1428.5714	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
137	5	03/2018	1428.5714	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
138	6	09/2018	1428.5714	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
139	7	03/2019	1428.5718	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
142	1	03/2017	2380.9524	SI	REPROGRAMADO	REPROGRAMACION CON 2 PAGOS HECHOS EN EL AÑO	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
153	2	09/2017	2390.9524	SI	REPROGRAMADO	REPROGRAMACION CON 2 PAGOS HECHOS EN EL AÑO	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA
154	3	03/2018	2380.9524	SI	REPROGRAMADO	REPROGRAMACION CON 2 PAGOS HECHOS EN EL AÑO	1	CARLOS NOLE	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA

Figura 42. Reporte de Programación de Pagos en formato PDF

Fuente: Elaborado por el Autor

ReporteProgramacionPagos.xls [Modo de compatibilidad] - Microsoft Excel

ARCHIVO INICIO INSERTAR DISEÑO DE PÁGINA FORMULAS DATOS REVISAR VISTA DESARROLLADOR EQUIPO

SansSerif 10 A A Ajustar texto General

Formato De formato Estilos de Insertar Eliminar Formato

Autosuma Rellenar Borrar Ordenar y filtrar Buscar y seleccionar

USUARIO: admin

SICA vs 1.0

	NRO. CUOTA	MES AÑO	ACTIVO	ESTADO	REFERENCIA	ACREEDOR	TIPO DEUDA	ESTADO ACREENCIA
10	132	1 03/2016	SI	PROGRAMADO		CARLOS NOLE	COMERCIAL	RECONOCIDA
15	134	2 09/2016	SI	PROGRAMADO		CARLOS NOLE	COMERCIAL	RECONOCIDA
20	135	3 03/2017	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	CARLOS NOLE	COMERCIAL	RECONOCIDA
25	136	4 09/2017	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	CARLOS NOLE	COMERCIAL	RECONOCIDA
30	137	5 03/2018	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	CARLOS NOLE	COMERCIAL	RECONOCIDA
35	138	6 09/2018	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	CARLOS NOLE	COMERCIAL	RECONOCIDA
40	139	7 03/2019	No	PROGRAMADO	SE REALIZO LA REPROGRAMACION DE PAGOS	CARLOS NOLE	COMERCIAL	RECONOCIDA
45	162	1 03/2017	SI	REPROGRAMADO	REPROGRAMACION CON 2 PAGOS HECHOS EN EL AÑO	CARLOS NOLE	COMERCIAL	RECONOCIDA
50	163	2 09/2017	SI	REPROGRAMADO	REPROGRAMACION CON 2 PAGOS HECHOS EN EL AÑO	CARLOS NOLE	COMERCIAL	RECONOCIDA
55	164	3 03/2018	SI	REPROGRAMADO	REPROGRAMACION CON 2 PAGOS HECHOS EN EL AÑO	CARLOS NOLE	COMERCIAL	RECONOCIDA

Figura 43. Reporte de Programación de Pagos en formato XLS
Fuente: Elaborado por el Autor

4.2.5.5. ACREENCIAS.

En esta interfaz se puede generar el Reporte de Acreencias y tendrá como criterios de búsqueda: Nro. Acreencia, acreedor, Nro. Documento, resolución, fecha de resolución (rango) y estado de acreencia (reconocida, no reconocida o todos).

Reporte: Acreencia

Criterio (*) Nro. Documento ▼

Exportar a Excel Exportar a PDF

(*) Campos obligatorios.

Figura 44. Interfaz para generar el Reporte de Acreencias
Fuente: Elaborado por el Autor

REPORTE: ACREENCIA

NRO. ACREENCIA	MUNICIPIO	NRO. DOCUMENTO	RESOLUCION	FECHA RESOLUCION	ID ACREEDOR	ACREEDOR	TIPO ACREEDOR	TIPO DEUDA	TIPO DEUDA	TIPO ACREENCIA	ESTADO ACREENCIA
2	20000	03214	Nº 0686-2015/SCO-INDECOP	13/03/2015	11	CAJA MUNICIPAL AHORRO Y CREDITO SULLANA S.A	PROVEEDOR	2	LABORAL	COMERCIALES	RECONOCIDA
1	10000	012345	0686-2014/SCO-INDECOP	12/03/2015	1	CARLOS MOLE	TRABAJADOR	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA

Figura 45. Reporte de Acreencia en formato PDF.
Fuente: Elaborado por el Autor

Reporte: Acreencias (Modo de compatibilidad) - Microsoft Excel

REPORTE: ACREENCIA

NRO. ACREENCIA	MUNICIPIO	NRO. DOCUMENTO	RESOLUCION	FECHA RESOLUCION	ID ACREEDOR	ACREEDOR	TIPO ACREEDOR	TIPO DEUDA	TIPO DEUDA	TIPO ACREENCIA	ESTADO ACREENCIA
2	20000	03214	Nº 0686-2015/SCO-INDECOP	13/03/2015	11	CAJA MUNICIPAL AHORRO Y CREDITO SULLANA S.A	PROVEEDOR	2	LABORAL	COMERCIALES	RECONOCIDA
1	10000	012345	0686-2014/SCO-INDECOP	12/03/2015	1	CARLOS MOLE	TRABAJADOR	1	COMERCIAL	PROCESOS CIVILES	RECONOCIDA

Figura 46. Reporte de Acreencia en formato XLS.
Fuente: Elaborado por el Autor

4.2.5.6. *ACREEDOR.*

En este submenú se encontrará a su vez los submenús para generar los reportes de trabajador y proveedor.

4.2.5.4.1. *TRABAJADOR.*

En esta interfaz se puede generar el Reporte de Trabajador y tendrá como criterios de búsqueda: DNI y Nombre y/o apellidos.

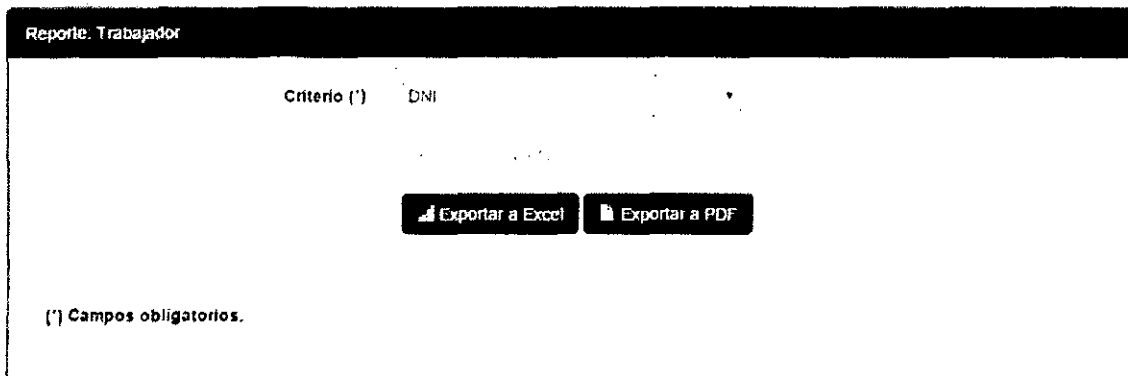


Figura 47. Interfaz para generar el Reporte de Trabajador
Fuente: Elaborado por el Autor

4.2.5.4.2. *PROVEEDOR.*

En esta interfaz se puede generar el Reporte de Proveedor y tendrá como criterios de búsqueda: RUC y razón social.

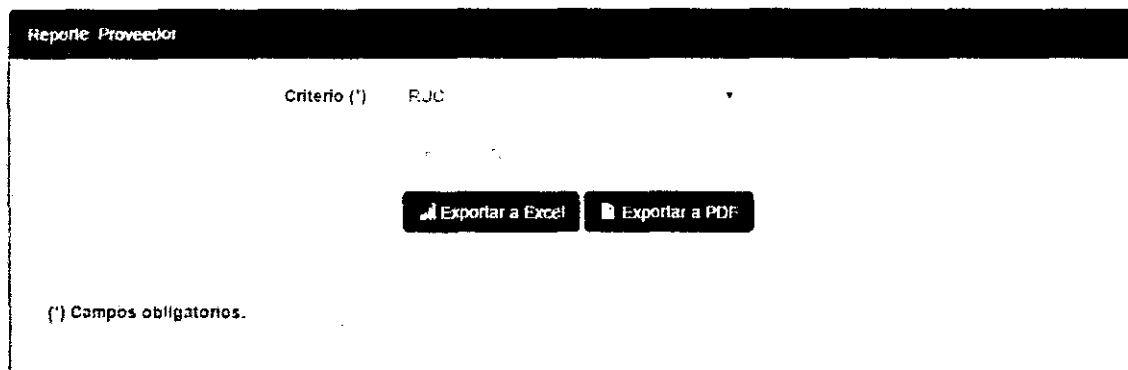


Figura 48. Interfaz para generar el Reporte de Proveedor
Fuente: Elaborado por el Autor

4.2.5.7. BENEFICIARIO.

En esta interfaz se puede generar el Reporte de Beneficiario y tendrá como criterios de búsqueda: Acreedor, DNI y nombre y/o apellidos.

Reporte: Beneficiario

Criterio (*) Acreedor ▼

Buscar

Exportar a Excel Exportar a PDF

(*) Campos obligatorios.

Figura 49. Interfaz para generar el Reporte de Beneficiario
Fuente: Elaborado por el Autor

CAPÍTULO 5: CONSTRUCCIÓN Y PRUEBAS

5.1. FRAMEWORKS

5.1.1. SPRING FRAMEWORK

Para desarrollar la arquitectura de Software de SICA se utilizó el framework **Spring** versión 3 que proporciona un modelo de programación y configuración completa para aplicaciones empresariales basadas en Java.

Las características que posee este poderoso framework es la Inyección de dependencias programación orientada aspectos incluyendo un administrador de transacciones, basado en un modelo MVC, soporta JDBC, JPA, etc.

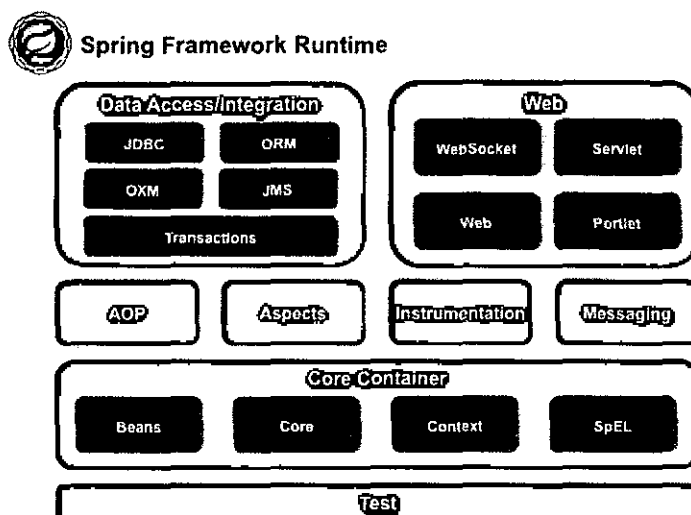


Figura 50. Overview of the Spring Framework

Fuente: <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/overview.html#overview-modules>

Spring maneja la siguiente estructura de archivos:

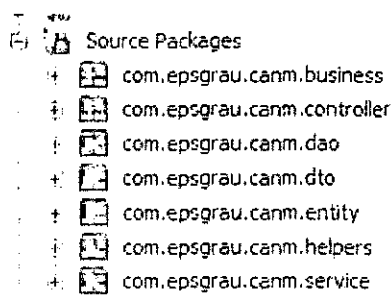


Figura 51. Estructura de Archivos SICA

Fuente: Elaborado por el Autor

En el paquete **entity** se encuentra todas las entidades en el cual se encuentra el mapeo de las tablas de nuestra base de datos. (**ANEXO 3**)

En el paquete **DAO (Data Access Object)** es donde se abstrae y encapsula el acceso a los datos (**ANEXO 4**). Se utiliza el **Hibernate Template** que es una plantilla para trabajar con Hibernate.

El paquete **business** es la capa de negocio donde se encuentra la lógica del negocio. (**ANEXO 5**)

El paquete **service** es la capa de servicio, permitiendo para implementación clases sea detectada de manera automática a través de la exploración de ruta de clases. (**ANEXO 6**)

En el paquete **controller** es donde se encuentra los controladores y con **@RequestMapping** se invoca un método con una petición http. (**ANEXO 7**)

En el paquete **DTO (Objeto de Transferencia de Datos)** esta objetos simples que no contienen lógica de negocio. (**ANEXO 8**)

Finalmente en **helpers** se encuentran clases que sirven de ayudante.

5.1.2. BOOTSTRAP

Para el diseño de las interfaces de SICA se utilizó el framework Bootstrap que hace que el desarrollo front-end web más rápido y fácil. Posee un Diseño Responsive lo que permite que SICA se visualice de manera correcta desde diferentes dispositivos. Provee de diferentes elementos HTML, componentes CSS y plugins JQuery impresionantes.

5.1.3. SPRING SECURITY

La autenticación y control de acceso de SICA se realizó con el framework Spring Security. Las características que posee es que: cuenta con protección contra ataques como la fijación de sesiones, clickjackijng, cross site solicitud falsificación, etc; integración con Spring Web MVC, integración API Servlet. (**ANEXO 11**)

5.2. HERRAMIENTAS DE SOFTWARE

5.2.1. JAVA

SICA se desarrolló con el lenguaje de programación **JAVA** por el lado del **back-end** porque es un lenguaje simple, orientado a objeto, distribuido, interpretado, sólido, seguro, de arquitectura neutral, portable, de alto desempeño, de multihilos y dinámico. Se utiliza JDK 1.7.

Simple porque tiene un sistema automático de asignación y liberación de memoria.

Orientado a objetos porque soporta técnicas de desarrollo OOP, reutilización de componente de software.

Distribuido porque contiene una gran biblioteca de clases para la utilización de protocolo TCP / IP, HTTP y FTP.

Interpretado porque el compilador Java traduce cada fichero fuente de clases a código de bytes, que puede ser interpretado por todas las máquinas.

Sólido debido que no se quiebra fácilmente ante errores de programación.

Seguro porque evita la corrupción de código evitan su manipulación.

Arquitectura neutral porque el compilador crea código de bytes que se envía al visualizador solicitado y se interpreta en la máquina que posee un intérprete de Java.

Portable debido a su arquitectura neutral.

Alto desempeño porque el resultado de la compilación resulta de una eficacia similar a las compilaciones de C y C++.

Multihilos porque puede realizarse varias cosas a la vez.

Dinámico porque permite nuevos métodos y variables en un objeto de biblioteca sin afectar a los objetos dependientes.

5.2.2. HTML

SICA usa el **HTML (Lenguaje de Marcas de Hipertexto)** para la elaboración de las páginas web. Cada etiqueta Html describe diferentes contenidos del documento (la estructura). Ahora SICA aprovecha los nuevos elementos de **HTML5**: web semántica, atributos de control, etc.

5.2.3. JAVASCRIPT

SICA utiliza el lenguaje de programación **JavaScript** por el lado del cliente para la parte de validaciones de formularios, llamadas **AJAX**, manipular **JSON**, etc.

JavaScript es el lenguaje de programación de Html y la Web.

NOTA: SICA utiliza jQuery que es una biblioteca JavaScript rápido, pequeño y rico en funciones. Hace las cosas como HTML el recorrido y manipulación

de documentos, manejo de eventos, animación, y Ajax mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores.

5.2.4. CSS

SICA utiliza la **CSS (Hoja de Estilos en Cascada)** para definir y crear la presentación de los documentos estructurados escritos en **HTML** (la presentación). **CSS** describe como los elementos deben presentarse en los diferentes medios.

5.2.5. iReport Designer

SICA utiliza **iReport** para la reportería. **iReport Designer 5.0.1** permite crear diseños muy sofisticados que contienen gráficos, imágenes, subinformes, tablas de contingencia, etc. Además permite el acceso a datos a través de **JDBC**, **TableModel**, **Java Beans**, **XML**, **Hibernate**, **CSV**, y las fuentes personalizadas. Luego de publicar sus informes en **PDF**, **RTF**, **XML**, **XLS**, **CSV**, **HTML**, **XHTML**, texto, **DOCX**, u **OpenOffice**.

5.2.6. Maven

El presente proyecto utiliza la herramienta de software **Maven** porque permite gestión, generación de informes y documentación de una pieza central de información y la construcción del mismo en el cual por medio del archivo **pom.xml (Project Object Model, ANEXO 9)** se encuentra la descripción del SICA, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos que requiere el SICA.

5.2.7. JSTL

SICA utiliza el **JavaSever Pages Standard Tag Libray (JSTL)** que es una colección de tags útiles de **JSP** que encapsulan la funcionalidad común de muchas aplicaciones **JSP**. **JSTL** tiene soporte para estructuras condicionales, iterativas, para manipular documentos **XML**, etiquetas de internalización, etiquetas **SQL**. También proporciona un marco de trabajo para la integración de etiquetas personalizadas con los tags **JSTL**.

5.2.8. Netbeans

Para el desarrollo de SICA se utilizó **Netbeans IDE 8.0.1** porque soporta las últimas tecnologías de java, se puede escribir código de manera rápida e inteligente (predictivo para autocompletar), permite administrar el proyecto de manera fácil y eficiente, permite diseñar la GUI de manera rápida y permite debuggar nuestro código. Además desde mi bachillerato lo he utilizado y su constante mejora me parece un buen editor para desarrollar aplicaciones en **Java**.

5.2.9. Log4j

En **SICA** se utiliza **Log4j** porque esta herramienta nos permite escribir mensajes de registro, cuyo propósito es dejar constancia de las transacciones en tiempo de ejecución que se efectúa en **SICA**. Se crearon (02) archivos con el nombre de **SicaController.log** (guarda las transacciones del paquete **Controller**) y **SicaDao.log** (guarda todas las transacciones del paquete **DAO**) en el cual se encontrara lo mencionado anteriormente y se encuentra ubicado en la carpeta logs de nuestro Apache Tomcat.

Se almacena (02) tipos de mensajes las tipo **INFO** (información) cuando un proceso se ejecutó correctamente y los tipo **ERROR** para que nos indique en que línea y el motivo que ha producido el error.

La configuración la encontramos en el archivo **log4j.properties** (**ANEXO 15**)

This PC > Local Disk (C:) > appServers > apache-tomcat-8.0.26 > logs

Name	Date modified	Type	Size
manager.2015-12-12.log	12-12-15 9:40 PM	Text Document	46 KB
manager.2015-12-13.log	13-12-15 10:13 PM	Text Document	23 KB
manager.2015-12-14.log	14-12-15 11:22 PM	Text Document	7 KB
manager.2015-12-19.log	19-12-15 6:31 PM	Text Document	58 KB
manager.2015-12-21.log	21-12-15 11:50 PM	Text Document	9 KB
manager.2015-12-22.log	22-12-15 2:06 PM	Text Document	15 KB
manager.2015-12-24.log	24-12-15 7:26 PM	Text Document	2 KB
manager.2015-12-25.log	25-12-15 11:56 PM	Text Document	105 KB
manager.2015-12-26.log	26-12-15 7:34 PM	Text Document	17 KB
manager.2015-12-27.log	27-12-15 12:55 PM	Text Document	6 KB
manager.2016-01-02.log	02-01-16 11:07 PM	Text Document	43 KB
manager.2016-01-04.log	04-01-16 11:49 PM	Text Document	10 KB
manager.2016-01-05.log	05-01-16 11:58 PM	Text Document	32 KB
manager.2016-01-06.log	06-01-16 11:59 PM	Text Document	12 KB
manager.2016-01-07.log	07-01-16 10:26 PM	Text Document	3 KB
manager.2016-01-08.log	08-01-16 11:11 PM	Text Document	2 KB
manager.2016-01-09.log	09-01-16 11:17 PM	Text Document	23 KB
manager.2016-01-10.log	10-01-16 11:24 PM	Text Document	31 KB
manager.2016-01-11.log	11-01-16 10:49 PM	Text Document	25 KB
manager.2016-01-12.log	12-01-16 9:48 PM	Text Document	3 KB
SicaController.log	12-01-16 10:47 PM	Text Document	562 KB
SicaDao.log	09-01-16 4:09 PM	Text Document	116 KB
springWebMaven.log	25-10-15 12:20 AM	Text Document	1 KB
springWebMavenController.log	25-10-15 12:18 AM	Text Document	2 KB

Figura 52. Logs del SICA
Fuente: Elaborado por el Autor

SicaController.log

2015-10-25 22:07:59,391 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-59]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:07:59,510 INFO	com.epsgrau.cana.controller.TipoDocumentoController	77	[http-apr-1234-exec-19]	com.epsgrau.cana.controller.TipoDocumentoController	: Name
2015-10-25 22:16:31,378 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-27]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:17:51,820 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-30]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:29:35,158 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-41]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:31:40,713 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-44]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:32:52,245 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-43]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:32:55,901 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-42]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:33:01,786 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-47]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:33:03,894 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-40]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:33:48,595 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-35]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:33:51,602 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-40]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:33:58,759 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-46]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:34:45,901 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-45]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:34:47,221 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-38]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:34:51,641 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-45]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:35:21,782 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-43]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:35:22,702 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-42]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:35:26,629 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-46]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:38:32,475 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-46]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:38:33,836 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-40]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:38:35,836 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-39]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:41:04,217 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-42]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:43:10,542 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-45]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:43:14,169 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-41]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:44:00,258 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-43]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:45:09,628 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-47]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:45:53,309 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-39]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:46:46,668 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-41]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 22:48:35,185 INFO	com.epsgrau.cana.controller.TipoDocumentoController	74	[http-apr-1234-exec-39]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 23:09:17,641 INFO	com.epsgrau.cana.controller.TipoDocumentoController	73	[http-apr-1234-exec-49]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 23:09:51,084 INFO	com.epsgrau.cana.controller.TipoDocumentoController	73	[http-apr-1234-exec-50]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 23:10:53,186 INFO	com.epsgrau.cana.controller.TipoDocumentoController	73	[http-apr-1234-exec-40]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 23:11:00,202 INFO	com.epsgrau.cana.controller.TipoDocumentoController	73	[http-apr-1234-exec-57]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 23:21:10,485 INFO	com.epsgrau.cana.controller.TipoDocumentoController	73	[http-apr-1234-exec-53]	com.epsgrau.cana.controller.TipoDocumentoController	: list
2015-10-25 23:22:36,980 INFO	com.epsgrau.cana.controller.TipoDocumentoController	73	[http-apr-1234-exec-53]	com.epsgrau.cana.controller.TipoDocumentoController	: list

Figura 53. Estructura de un archivo log de SICA
Fuente: Elaborado por el Autor

5.3. PATRONES DE DISEÑO

Al hacer uso de un framework estamos usando un conjunto de patrones de diseño que la componen, en este caso mencionaremos algunas para que el lector se familiarice con SICA.

5.3.1. MODELO – VISTA – CONTROLADOR

Spring Framework proporciona el **Modelo – Vista – Controlador** lo que permite que SICA los diferentes aspectos de la aplicación estén separados (los datos, la lógica del negocio y la interfaz de usuario).

El **Modelo** encapsula los datos de la aplicación y en general consiste en POJO.

La **Vista** es responsable de mostrar el modelo de datos y por lo general la salida con **Html** que el navegador del cliente puede interpretar.

El **Controlador** es responsable para procesar las peticiones de los usuarios y construir apropiadamente el **Modelo** y pasar esto a la **Vista** para que se muestre.

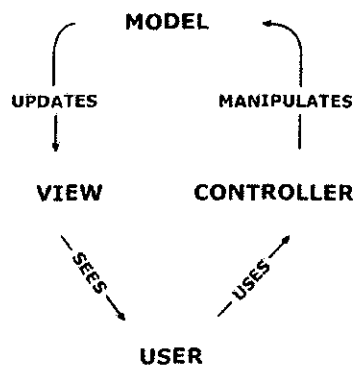


Figura 54. Una típica colaboración entre los componentes de un MVC
Fuente: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

Luego de explicar de manera general el M-V-C, ahora en Spring está diseñado alrededor del DispatcherServlet:

- Después de recibir una petición HTTP, el DispatcherServlet consulta el HandlerMapping para llamar al Controlador apropiado.
- El Controlador toma la petición y llama el método de servicio apropiado base en GET o POST. Este servicio establece el modelo de datos basado en la lógica de negocio definido y retorna el nombre de la vista al DispatcherServlet.
- El DispatcherServlet (**ANEXO 10**) se ayuda de ViewResolver y recoge la vista definida para la petición.
- Cada vista está finalizada, El DispatcherServlet pasa el modelo de datos a la vista que finalmente se muestra en el navegador.

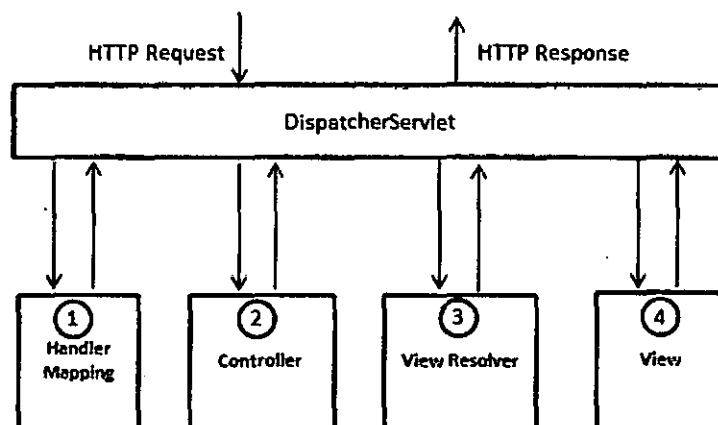


Figura 55. The DispatcherServlet

Fuente: http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm

Los componentes mencionados anteriormente (HandlerMapping, Controlador y viewResolver) son partes del WebApplicationContext que es una extensión del ApplicationContext con las características necesarias para las aplicaciones Web (**ANEXO 11**).

5.3.2. INYECCIÓN DE DEPENDENCIAS

Spring framework mediante su contenedor de inversión de control permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java, y lo lleva a cabo a través de la inyección de dependencias (permite que las clases mantengan su independencia).

5.4. PRUEBAS DEL SICA

A continuación se mostrarán las pruebas de funcionalidad y carga que se realizaron en el SICA que tomando como referencia la ya mencionada ISO / IEC 2500 se busca una buena calidad¹¹ de software y debe cumplir con las siguientes características:

FUNCIONALIDAD

- **Complejidad funcionalidad.** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.

EFICIENCIA

- **Comportamiento temporal.**

¹¹ Según la IEEE: Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas.

- **Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el SICA lleva a cabo su función bajo condiciones determinadas.
- **Capacidad.** Grado en que los límites máximos de SICA cumple con los requisitos.

5.4.1. PRUEBAS DE FUNCIONALIDAD

PROCESO	GENERAR PROGRAMACIÓN DE PAGOS
DESCRIPCION	Proceso en la clase ProgramacionPagoController, el cual recibe los valores desde registrarProgramacionPagos.js. El orden de ingreso de los datos es consecutivo, en primer lugar el formulario debe estar desactivado luego hacer click en botón Nuevo para que se active el formulario y se continúe con el ingreso de un Nro. Acreencia activo, campos Nro. Acreencia y Nro. Cuotas deben ser tipo numérico. Los campos Nro. Acreencia, Importe, Nro. Cutas y Desde deben ser campos obligatorios; el usuario puede cancelar el proceso haciendo click en el botón Cancelar en el momento que desee; si existiese un error se le notificará al usuario. En caso de acierto se le comunica al usuario, terminando el proceso.
PRUEBAS DE FUNCIONALIDAD	
PROCEDIMIENTO: USO DE SOFTWARE	
	ID: PF001
	PRUEBA: Generar Programación de Pagos
PROCEDIMIENTO I	ACCIÓN: Generar la Programación de Pagos en el orden lógico establecido.
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna.
	ID: PF002
	PRUEBA: Generar Programación de Pagos
PROCEDIMIENTO II	ACCIÓN: Validar Nro. Acreencia si existe
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna
	ID: PF003
	PRUEBA: Generar Programación de Pagos
PROCEDIMIENTO III	ACCIÓN: Validar que no se genere programación de pagos más de (01) vez.
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna
	ID: PF004
	PRUEBA: Generar Programación de Pagos
PROCEDIMIENTO IV	ACCIÓN: Validar campos numéricos
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna
	ID: PF005
	PRUEBA: Generar Programación de Pagos
PROCEDIMIENTO V	ACCIÓN: Validar campos obligatorios
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna

Tabla 45. Documento de Trabajo Funcionalidad: Programación de Pagos

Fuente: Távara, A. (2015). Elaboración de Documentos de Trabajo y uso de Herramientas de Software para el proceso de Testing de Aplicaciones Web

PROCESO	
GENERAR REPROGRAMACIÓN DE PAGOS	
DESCRIPCION	Proceso en la clase ProgramacionPagoController, el cual recibe los valores desde registrarReprogramacionPagos.js. El orden de ingreso de los datos es consecutivo, en primer lugar el formulario debe estar desactivado luego hacer click en botón Nuevo para que se active el formulario y se continúe con el ingreso de un Nro. Acreencia activo, campos Nro. Acreencia y Nro. Cuotas deben ser tipo numérico. Los campos Nro. Acreencia, Importe, Nro. Cutas, Desde y Referencia deben ser campos obligatorios; el usuario puede cancelar el proceso haciendo click en el botón Cancelar en el momento que desee; si existiese un error se le notificará al usuario. En caso de acierto se le comunica al usuario, terminando el proceso.
PRUEBAS DE FUNCIONALIDAD	
PROCEDIMIENTO: USO DE SOFTWARE	
	ID: PF001
	PRUEBA: Generar Reprogramación de Pagos
PROCEDIMIENTO I	ACCIÓN: Generar la Reprogramación de Pagos en el orden lógico establecido.
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna.
	ID: PF002
	PRUEBA: Generar Reprogramación de Pagos
PROCEDIMIENTO II	ACCIÓN: Validar Nro. Acreencia si existe
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna
	ID: PF003
	PRUEBA: Generar Reprogramación de Pagos
PROCEDIMIENTO III	ACCIÓN: Validar que no se genere la reprogramación de pagos más de (01) vez.
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna
	ID: PF004
	PRUEBA: Generar Reprogramación de Pagos
PROCEDIMIENTO IV	ACCIÓN: Validar campos numéricos
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna
	ID: PF005
	PRUEBA: Generar Reprogramación de Pagos
PROCEDIMIENTO V	ACCIÓN: Validar campos obligatorios
	RESULTADO: Satisfactorio
	OBSERVACIONES: Ninguna

Tabla 46. Documento de Trabajo Funcionalidad: Reprogramación de Pagos

Fuente: Távara, A. (2015). Elaboración de Documentos de Trabajo y uso de Herramientas de Software para el proceso de Testing de Aplicaciones Web

PROCESO	
REALIZAR PAGOS SEMESTRALES	
DESCRIPCION	Proceso en la clase PagoController, el cual recibe los valores desde registrarPagosRealizados.js. El orden de ingreso de los datos es consecutivo, en primer lugar el formulario debe estar desactivado luego hacer click en botón Nuevo para que se active el formulario y se continúe con el ingreso de un Nro. Acreencia activo y Mes / Año (Fecha de cuota programada), campos Nro. Acreencia debe ser tipo numérico. Los campos Nro. Acreencia, Nro. Programación Pago, Acreedor, Monto, Fecha Pago y Tipo de Pago deben ser campos obligatorios; validar que no se realice

dobles pago de una cuota; el usuario puede cancelar el proceso haciendo click en el botón Cancelar en el momento que desee; si existiese un error se le notificará al usuario. En caso de acierto se le comunica al usuario, terminando el proceso.

PRUEBAS DE FUNCIONALIDAD

PROCEDIMIENTO: USO DE SOFTWARE

	ID: PF001
PROCEDIMIENTO I	PRUEBA: Realizar Pagos Semestrales ACCIÓN: Realizar un pago semestral. RESULTADO: Satisfactorio OBSERVACIONES: Ninguno
	ID: PF002
PROCEDIMIENTO II	PRUEBA: Realizar Pagos Semestrales ACCIÓN: Validar que no se realice doble pago de una cuota RESULTADO: Satisfactorio OBSERVACIONES: Ninguna
	ID: PF003
PROCEDIMIENTO III	PRUEBA: Realizar Pagos Semestrales ACCIÓN: Validar que cuota exista RESULTADO: Satisfactorio OBSERVACIONES: Ninguna
	ID: PF004
PROCEDIMIENTO IV	PRUEBA: Realizar Pagos Semestrales ACCIÓN: Validar campos numéricos RESULTADO: Satisfactorio OBSERVACIONES: Ninguna
	ID: PF005
PROCEDIMIENTO V	PRUEBA: Realizar Pagos Semestrales ACCIÓN: Validar campos obligatorios RESULTADO: Satisfactorio OBSERVACIONES: Ninguna

Tabla 47. Documento de Trabajo Funcionalidad: Proceso de Pagos

Fuente: Távara, A. (2015). Elaboración de Documentos de Trabajo y uso de Herramientas de Software para el proceso de Testing de Aplicaciones Web

5.4.2. PRUEBAS DE RENDIMIENTO

Para realizar estas pruebas se usó la herramienta de software **Apache JMeter** 2.13 r diseñado para pruebas funcionales de comportamiento y para medir el rendimiento de recursos estáticos como dinámicos (Servicios Web: SOAP / REST), lenguajes dinámicos: PHP, Java, ASP .NET, objetos Java, base de datos, servidores FTP. Soporta diferentes tipos de carga.

En primer lugar debemos configurar la cantidad de hilos que es la cantidad de usuarios que simularan el acceso a la petición (10 usuarios que repetirán 10 veces la acción obteniendo una muestra total de 100 y que realizan la petición cada segundo) (**Figura 57**), configurar el ambiente lo mas real posible con los datos del servidor (IP, Puerto, usuario y contraseña de tomcat-users.xml en

Reg. 6632 — 4/5/16 UNP

nuestro caso, etc.) **(Figura 58)** y finalmente crear peticiones http en donde se configura la ruta que queremos probar del SICA.

En la siguiente imagen se muestra el Plan de Pruebas para SICA donde se ha escogido (03) procesos críticos (obviar el login):

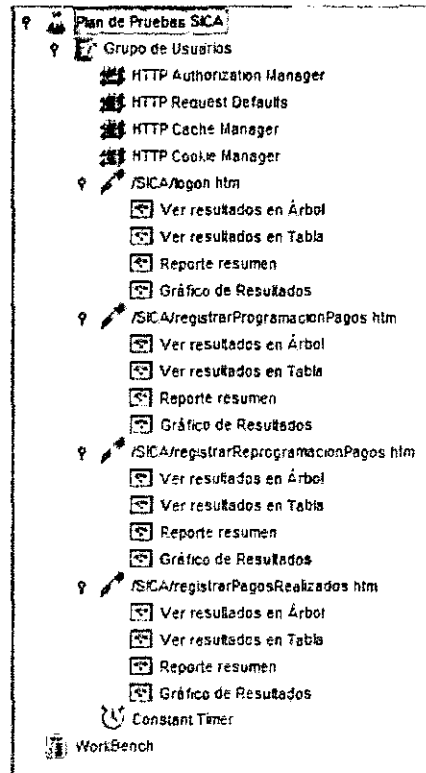


Figura 56. Plan de Pruebas SICA para medir el rendimiento
Fuente: Elaborado por el Autor

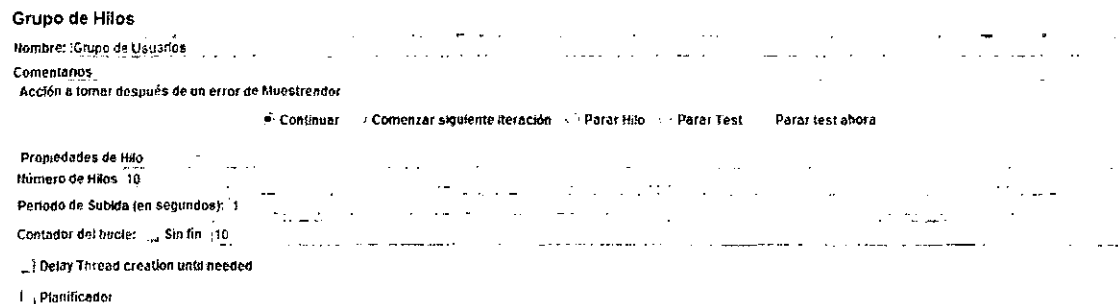


Figura 57. Configuración de Hilos
Fuente: Elaborado por el Autor

Gestor de Autorización HTTP

Nombre: HTTP Authorization Manager

Comentarios

Options

☐ Clear auth on each iteration?

Autorizaciones Almacenadas en el Gestor de Autorización

URL Base	Nombre de Usuario	Contraseña	Dominio	Dominio (realm)	Mechanism
	admin	*****			BASIC_DIGEST

Figura 58. Configuración del Gestor de Autorización HTTP
Fuente: Elaborado por el Autor

Valores por Defecto para Petición HTTP

Nombre: HTTP Request Default

Comentarios

Servidor Web

Nombre de Servidor o IP: localhost Puerto: 1234 Timeout (milisegundos)

Petición HTTP

Implementación HTTP: Protocolo: Codificación del contenido:

Ruta:

Parámetros

Enviar Parámetros Con la Petición:

Nombre	Valor	¿Codificar?	¿Incluir Equivalencia?

Detalles Añadir Add from Clipboard Up Down

Servidor Proxy

Nombre de Servidor o IP: Puerto: Nombre de Usuario: Contraseña:

Embedded Resources from HTML files

Recuperar Todos los Recursos Embebidos de Archivos HTML

Las URLs embebidas deben coincidir a:

Figura 59. Configuración para los valores por defecto de las peticiones HTTP.
Fuente: Elaborado por el Autor

Los resultado de estas pruebas serán discutidas en el siguiente capítulo.

5.4.4.1. PROGRAMACIÓN DE PAGOS.

Luego de crear una petición http se debe configurar la ruta donde se ejecuta nuestra petición **/SICA/generarProgramacionPagos.htm**, que tipo de método utilizar en nuestro caso **POST** y agregar los parámetros con su respectivo valor: **nroAcreencia**, **importe**, **nroCuotas**, **desde**, **referencia**. La ruta es la que se mapea en nuestro contralador (**ANEXO 16**).

Petición HTTP

Nombre: /SICA/RegistrarProgramacionPagos.htm

Comentarios

Servidor Web: Puerto: 1234 Timeout (milisegundos): Conexión: Respuesta:

Nombre de Servidor o IP: localhost

Petición HTTP

Implementación HTTP: Protocolo: Método: POST Codificación del contenido:

Ruta: /SICA/generarProgramacionPagos.htm

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST ☐ Cabeceras compatibles con navegadores

Parameters Body Data

Enviar Parámetros Con la Petición:

Nombre	Valor	¿Codificar?	¿Incluir Equals?
nroAcreencia	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
importe	10000	<input type="checkbox"/>	<input checked="" type="checkbox"/>
nroCuotas	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail Añadir Add from Clipboard Borrar Up Down

Enviar un archivo Con la Petición

Nombre de Archivo	Nombre de Parámetro	Tipo MIME
<input type="text"/>	<input type="text"/>	<input type="text"/>

Añadir Navegar... Borrar

Figura 60. Configuración para realizar la petición de Programación de Pagos
Fuente: Elaborado por el Autor

5.4.4.2. REPROGRAMACIÓN DE PAGOS.

Luego de crear una petición http se debe configurar la ruta donde se ejecuta nuestra petición /SICA/regarstrarReprogramacionPagos.htm, que tipo de método utilizar en nuestro caso POST y agregar los parámetros con su respectivo valor: nroAcreencia, importe, nroCuotas, desde, referencia. La ruta es la que se mapea en nuestro contralador **(ANEXO 16)**.

Petición HTTP

Nombre: /SICA/regarstrarReprogramacionPagos.htm

Comentarios

Servidor Web: Puerto: 1234 Timeout (milisegundos): Conexión: Respuesta:

Nombre de Servidor o IP: localhost

Petición HTTP

Implementación HTTP: Protocolo: Método: POST Codificación del contenido:

Ruta: /SICA/generarReprogramacionPagos.htm

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST ☐ Cabeceras compatibles con navegadores

Parameters Body Data

Enviar Parámetros Con la Petición:

Nombre	Valor	¿Codificar?	¿Incluir Equals?
nroAcreencia	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
importe	7152.9572	<input type="checkbox"/>	<input checked="" type="checkbox"/>
nroCuotas	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail Añadir Add from Clipboard Borrar Up Down

Enviar un archivo Con la Petición

Nombre de Archivo	Nombre de Parámetro	Tipo MIME
<input type="text"/>	<input type="text"/>	<input type="text"/>

Añadir Navegar... Borrar

Figura 61. Configuración para realizar la petición de Reprogramación de Pagos
Fuente: Elaborado por el Autor

5.4.4.3. PROCESO DE PAGOS.

Luego de crear una petición http se debe configurar la ruta donde se ejecuta nuestra petición **/SICA/regarPagosRealizados.htm**, que tipo de método utilizar en nuestro caso **POST** y agregar los parámetros con su respectivo valor: **idProgPag**, **idAcreedor**, **idBeneficiario**, **fechaPago**, **monto**, **referencia**, **tipoPago**, **fecha**. La ruta es la que se mapea en nuestro contralador (0).

Petición HTTP

Nombre: **/SICA/regarPagosRealizados.htm**

Comentarios

Servidor Web: **localhost** Puerto: **1234** Timeout (milisegundos): Conexión: Respuesta:

Petición HTTP

Implementación HTTP: Protocolo: Método: **POST** Codificación del contenido:

Ruta: **/SICA/regarPagosRealizados.htm**

☐ Redirigir Automáticamente ☒ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Usar 'multipart/form-data' para HTTP POST ☐ Cabeceras compatibles con navegadores

Parámetros Body Data

Enviar Parámetros Con la Petición:

Nombre	Valor	¿Codificar?	¿Incluir Equals?
idProgPag	162	<input type="checkbox"/>	<input checked="" type="checkbox"/>
idAcreedor	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
idBeneficiario	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail Añadir Add from Clipboard Borrar Up Down

Enviar un archivo Con la Petición

Nombre de Archivo	Nombre de Parámetro	Tipo MIME

Añadir Navegar Borrar

Figura 62. Configuración para realizar la petición de Pagos

Fuente: Elaborado por el Autor

CAPÍTULO 6: PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS

6.1. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

6.1.1. PROGRAMACIÓN DE PAGOS

Se verifica que para la Programación de Pagos una muestra de 100 realizado por 10 usuarios el SICA responde correctamente donde se obtiene 0% de error con un rendimiento 1.3 seg. A continuación se muestran los diferentes tipos de reportes que nos proporciona la herramienta JMeter en el cual podemos ver y analizar algún otro campo. Los resultados son expresados en milisegundos.

6.1.1.1. RESULTADOS EN TABLA

Muestra #	Tiempo de comienzo	Nombre de hilo	Etiqueta	Tiempo de Muestra	Estado	Bytes	Latencia	Tiempo Conectado (ms)
1	11:56:13 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	243	Success	232	242	0
2	11:56:13 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
3	11:56:13 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	15	Success	256	14	0
4	11:56:13 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	15	Success	256	13	0
5	11:56:14 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
6	11:56:14 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
7	11:56:14 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
8	11:56:14 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
9	11:56:14 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	15	Success	256	14	0
10	11:56:14 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
11	11:56:22 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	108	Success	232	107	0
12	11:56:22 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
13	11:56:22 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0

14	11:56:22 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	18	Success	256	17	0
15	11:56:22 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	20	Success	256	18	0
16	11:56:22 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	20	Success	256	19	0
17	11:56:22 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	21	Success	256	21	0
18	11:56:22 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
19	11:56:22 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
20	11:56:23 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
21	11:56:31 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
22	11:56:31 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
23	11:56:31 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	16	Success	256	15	0
24	11:56:31 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
25	11:56:31 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
26	11:56:31 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
27	11:56:31 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
28	11:56:31 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	11	Success	256	10	0
29	11:56:31 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	11	Success	256	10	0
30	11:56:31 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
31	11:56:39 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
32	11:56:39 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	7	Success	256	6	0
33	11:56:39 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
34	11:56:39 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	5	Success	256	5	0
35	11:56:39 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
36	11:56:40 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	6	Success	256	6	0
37	11:56:40 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
38	11:56:40 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	5	Success	256	5	0
39	11:56:40 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
40	11:56:40 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	7	Success	256	6	0
41	11:56:48 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	17	Success	256	16	0

42	11:56:48 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	7	Success	256	6	0
43	11:56:48 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
44	11:56:48 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
45	11:56:48 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
46	11:56:48 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
47	11:56:48 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
48	11:56:48 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	15	Success	256	14	0
49	11:56:48 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	17	Success	256	15	0
50	11:56:49 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
51	11:56:56 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	6	Success	256	6	0
52	11:56:57 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
53	11:56:57 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
54	11:56:57 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
55	11:56:57 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	14	Success	256	12	0
56	11:56:57 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
57	11:56:57 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	14	Success	256	12	0
58	11:56:57 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
59	11:56:57 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
60	11:56:57 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	15	Success	256	14	0
61	11:57:05 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	17	Success	256	15	0
62	11:57:05 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
63	11:57:05 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
64	11:57:05 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
65	11:57:05 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
66	11:57:05 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
67	11:57:05 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	12	Success	256	11	0
68	11:57:06 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
69	11:57:06 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	16	Success	256	15	0

70	11:57:06 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
71	11:57:13 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
72	11:57:14 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
73	11:57:14 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	15	Success	256	14	0
74	11:57:14 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
75	11:57:14 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	11	Success	256	10	0
76	11:57:14 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	12	Success	256	11	0
77	11:57:14 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	17	Success	256	15	0
78	11:57:14 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	18	Success	256	16	0
79	11:57:14 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	7	Success	256	7	0
80	11:57:14 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
81	11:57:22 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	20	Success	256	18	0
82	11:57:22 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	8	Success	256	8	0
83	11:57:22 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
84	11:57:23 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
85	11:57:23 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
86	11:57:23 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
87	11:57:23 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	9	Success	256	9	0
88	11:57:23 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	8	Success	256	7	0
89	11:57:23 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	9	Success	256	7	0
90	11:57:23 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
91	11:57:31 PM	Grupo de Usuarios 1-6	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
92	11:57:31 PM	Grupo de Usuarios 1-8	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
93	11:57:31 PM	Grupo de Usuarios 1-5	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
94	11:57:31 PM	Grupo de Usuarios 1-7	/SICA/registrarProgramacionPagos.htm	6	Success	256	5	0
95	11:57:32 PM	Grupo de Usuarios 1-2	/SICA/registrarProgramacionPagos.htm	16	Success	256	14	0
96	11:57:32 PM	Grupo de Usuarios 1-3	/SICA/registrarProgramacionPagos.htm	9	Success	256	8	0
97	11:57:32 PM	Grupo de Usuarios 1-9	/SICA/registrarProgramacionPagos.htm	5	Success	256	5	0

98	11:57:32 PM	Grupo de Usuarios 1-4	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
99	11:57:32 PM	Grupo de Usuarios 1-10	/SICA/registrarProgramacionPagos.htm	10	Success	256	9	0
100	11:57:32 PM	Grupo de Usuarios 1-1	/SICA/registrarProgramacionPagos.htm	13	Success	256	12	0

Tabla 48. Resultados en Tabla para la Programación de Pagos

Fuente: Elaborado por el Autor

6.1.1.2. REPORTE RESUMEN

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/SICA/registrarProgramacionPagos.htm	100	14	5	243	25.29	0.00%	1.3	0.32	255.5
Total	100	14	5	243	25.29	0.00%	1.3	0.32	255.5

Tabla 49. Reporte Resumen para la Programacion de Pagos

Fuente: Elaborado por el Autor

6.1.2. REPROGRAMACION DE PAGOS

Se verifica que para la Reprogramación de Pagos una muestra de 100 realizado por 10 usuarios el SICA responde correctamente donde se obtiene 0% de error con un rendimiento de 1.3 seg. A continuación se muestran los diferentes tipos de reportes que nos proporciona la herramienta JMeter en el cual podemos ver y analizar algún otro campo. Los resultados son expresados en milisegundos.

6.1.2.1. RESULTADOS EN TABLA

Muestra #	Tiempo de comienzo	Nombre de hilo	Etiqueta	Tiempo de Muestra	Estado	Bytes	Latencia	Tiempo Conectado (ms)
1	11:56:15 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	230	Success	258	229	0
2	11:56:15 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	51	Success	258	51	0
3	11:56:15 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	222	Success	232	221	0
4	11:56:15 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	15	Success	258	13	0

5	11:56:16 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
6	11:56:16 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	5	Success	258	5	0
7	11:56:16 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	7	Success	258	6	0
8	11:56:16 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	15	Success	258	13	0
9	11:56:16 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
10	11:56:16 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
11	11:56:24 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	11	Success	258	10	0
12	11:56:24 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
13	11:56:24 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	9	Success	258	8	0
14	11:56:24 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	12	Success	258	10	0
15	11:56:24 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	13	Success	258	12	0
16	11:56:24 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	14	Success	258	12	0
17	11:56:24 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	14	Success	258	13	0
18	11:56:24 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	15	Success	258	14	0
19	11:56:24 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	9	Success	258	8	0
20	11:56:25 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	6	Success	258	6	0
21	11:56:33 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	9	Success	258	9	0
22	11:56:33 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
23	11:56:33 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	7	Success	258	6	0
24	11:56:33 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
25	11:56:33 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
26	11:56:33 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	19	Success	258	17	0
27	11:56:33 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	20	Success	258	17	0
28	11:56:33 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	21	Success	258	19	0
29	11:56:33 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	21	Success	258	19	0
30	11:56:33 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	18	Success	258	15	0
31	11:56:41 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	7	Success	258	7	0
32	11:56:41 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	11	Success	258	10	0

33	11:56:41 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	12	Success	258	11	0
34	11:56:41 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
35	11:56:41 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	10	Success	258	9	0
36	11:56:42 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	5	Success	258	5	0
37	11:56:42 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	5	Success	258	4	0
38	11:56:42 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
39	11:56:42 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
40	11:56:42 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
41	11:56:50 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
42	11:56:50 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
43	11:56:50 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	6	Success	258	6	0
44	11:56:50 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	5	Success	258	5	0
45	11:56:50 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	12	Success	258	11	0
46	11:56:50 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	13	Success	258	12	0
47	11:56:50 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
48	11:56:50 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	5	Success	258	5	0
49	11:56:50 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
50	11:56:51 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	7	Success	258	5	0
51	11:56:59 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	7	Success	258	6	0
52	11:56:59 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	5	Success	258	5	0
53	11:56:59 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	5	Success	258	5	0
54	11:56:59 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	6	Success	258	6	0
55	11:56:59 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	6	Success	258	6	0
56	11:56:59 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
57	11:56:59 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	13	Success	258	11	0
58	11:56:59 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	13	Success	258	12	0
59	11:56:59 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	13	Success	258	11	0
60	11:56:59 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0

61	11:57:07 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	10	Success	258	9	0
62	11:57:07 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	7	Success	258	7	0
63	11:57:07 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
64	11:57:07 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
65	11:57:07 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	14	Success	258	13	0
66	11:57:07 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	15	Success	258	13	0
67	11:57:07 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	15	Success	258	14	0
68	11:57:08 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
69	11:57:08 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
70	11:57:08 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	6	Success	258	6	0
71	11:57:16 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	9	Success	258	8	0
72	11:57:16 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	13	Success	258	11	0
73	11:57:16 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	9	Success	258	8	0
74	11:57:16 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
75	11:57:16 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
76	11:57:16 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	8	Success	258	8	0
77	11:57:16 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
78	11:57:16 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
79	11:57:16 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	15	Success	258	13	0
80	11:57:16 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	10	Success	258	9	0
81	11:57:24 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	17	Success	258	16	0
82	11:57:24 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
83	11:57:24 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	15	Success	258	13	0
84	11:57:25 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	9	Success	258	8	0
85	11:57:25 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	15	Success	258	14	0
86	11:57:25 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	18	Success	258	16	0
87	11:57:25 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	23	Success	258	21	0
88	11:57:25 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	18	Success	258	16	0

89	11:57:25 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	19	Success	258	16	0
90	11:57:25 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
91	11:57:33 PM	Grupo de Usuarios 1-6	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
92	11:57:33 PM	Grupo de Usuarios 1-8	/SICA/registrarReprogramacionPagos.htm	15	Success	258	14	0
93	11:57:33 PM	Grupo de Usuarios 1-5	/SICA/registrarReprogramacionPagos.htm	16	Success	258	14	0
94	11:57:33 PM	Grupo de Usuarios 1-7	/SICA/registrarReprogramacionPagos.htm	15	Success	258	14	0
95	11:57:34 PM	Grupo de Usuarios 1-2	/SICA/registrarReprogramacionPagos.htm	10	Success	258	8	0
96	11:57:34 PM	Grupo de Usuarios 1-3	/SICA/registrarReprogramacionPagos.htm	9	Success	258	9	0
97	11:57:34 PM	Grupo de Usuarios 1-9	/SICA/registrarReprogramacionPagos.htm	6	Success	258	5	0
98	11:57:34 PM	Grupo de Usuarios 1-4	/SICA/registrarReprogramacionPagos.htm	7	Success	258	6	0
99	11:57:34 PM	Grupo de Usuarios 1-10	/SICA/registrarReprogramacionPagos.htm	8	Success	258	7	0
100	11:57:34 PM	Grupo de Usuarios 1-1	/SICA/registrarReprogramacionPagos.htm	7	Success	258	6	0

Tabla 50. Resultados en Tabla para la Reprogramación de Pagos
Fuente: Elaborado por el Autor

6.1.2.2. REPORTE RESUMEN

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/SICA/registrarReprogramacionPagos.htm	100	15	5	230	30.69	0.00%	1.3	0.32	257.7
Total	100	15	5	230	30.69	0.00%	1.3	0.32	257.7

Tabla 51. Reporte Resumen para la Reprogramacion de Pagos
Fuente: Elaborado por el Autor

6.1.3. PROCESO DE PAGOS

Se verifica que para el Proceso de Pagos una muestra de 100 realizado por 10 usuarios el SICA responde correctamente donde se obtiene 0% de error con un rendimiento 1.3 seg. A continuación se muestran los diferentes tipos de reportes que nos proporciona la herramienta JMeter en el cual podemos ver y analizar algún otro campo. Los resultados son expresados en milisegundos.

6.1.3.1. RESULTADOS EN TABLA

Muestra #	Tiempo de comlenzo	Nombre de hilo	Etiqueta	Tiempo de Muestra	Estado	Bytes	Latencia	Tiempo Conectado (ms)
1	11:56:17 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	147	Success	232	145	0
2	11:56:17 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	187	Success	232	185	0
3	11:56:18 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	186	Success	232	184	0
4	11:56:18 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	165	Success	232	164	0
5	11:56:18 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	213	Success	232	212	0
6	11:56:17 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	355	Success	232	354	0
7	11:56:17 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	325	Success	232	324	0
8	11:56:18 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	51	Success	232	49	0
9	11:56:18 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	76	Success	232	74	0
10	11:56:18 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	55	Success	232	55	0
11	11:56:26 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	117	Success	232	115	0
12	11:56:26 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	159	Success	232	157	0
13	11:56:26 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	171	Success	232	169	0
14	11:56:26 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	161	Success	232	160	0
15	11:56:26 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	209	Success	232	207	0
16	11:56:26 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	211	Success	232	209	0
17	11:56:26 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	228	Success	232	226	0
18	11:56:26 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	213	Success	232	211	0

19	11:56:26 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	165	Success	232	165	0
20	11:56:27 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	112	Success	232	111	0
21	11:56:35 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	86	Success	232	84	0
22	11:56:35 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	90	Success	232	88	0
23	11:56:35 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	86	Success	232	85	0
24	11:56:35 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	198	Success	232	197	0
25	11:56:35 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	131	Success	232	130	0
26	11:56:35 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	80	Success	232	79	0
27	11:56:35 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	168	Success	232	167	0
28	11:56:35 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	169	Success	232	168	0
29	11:56:35 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	170	Success	232	169	0
30	11:56:35 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	170	Success	232	170	0
31	11:56:43 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	91	Success	232	90	0
32	11:56:43 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	47	Success	232	47	0
33	11:56:43 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	92	Success	232	92	0
34	11:56:43 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	54	Success	232	54	0
35	11:56:44 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	191	Success	232	190	0
36	11:56:44 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	110	Success	232	109	0
37	11:56:44 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	122	Success	232	121	0
38	11:56:44 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	77	Success	232	75	0
39	11:56:44 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	123	Success	232	122	0
40	11:56:44 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	146	Success	232	145	0
41	11:56:52 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	140	Success	232	140	0
42	11:56:52 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	145	Success	232	144	0
43	11:56:52 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	89	Success	232	87	0
44	11:56:52 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	134	Success	232	133	0
45	11:56:52 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	127	Success	232	126	0
46	11:56:52 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	187	Success	232	187	0

47	11:56:52 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	91	Success	232	90	0
48	11:56:52 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	125	Success	232	124	0
49	11:56:52 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	229	Success	232	227	0
50	11:56:53 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	86	Success	232	85	0
51	11:57:01 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	88	Success	232	87	0
52	11:57:01 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	105	Success	232	105	0
53	11:57:01 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	90	Success	232	89	0
54	11:57:01 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	85	Success	232	85	0
55	11:57:01 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	96	Success	232	93	0
56	11:57:01 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	141	Success	232	139	0
57	11:57:01 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	237	Success	232	236	0
58	11:57:01 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	141	Success	232	140	0
59	11:57:01 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	234	Success	232	233	0
60	11:57:01 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	73	Success	232	72	0
61	11:57:09 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	90	Success	232	89	0
62	11:57:09 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	199	Success	232	197	0
63	11:57:09 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	295	Success	232	294	0
64	11:57:09 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	233	Success	232	232	0
65	11:57:09 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	335	Success	232	334	0
66	11:57:09 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	246	Success	232	244	0
67	11:57:09 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	246	Success	232	244	0
68	11:57:10 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	63	Success	232	61	0
69	11:57:10 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	158	Success	232	157	0
70	11:57:10 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	162	Success	232	160	0
71	11:57:18 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	129	Success	232	128	0
72	11:57:18 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	177	Success	232	175	0
73	11:57:18 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	397	Success	232	396	0
74	11:57:18 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	426	Success	232	424	0

75	11:57:18 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	112	Success	232	111	0
76	11:57:18 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	214	Success	232	212	0
77	11:57:18 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	261	Success	232	260	0
78	11:57:18 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	403	Success	232	402	0
79	11:57:18 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	201	Success	232	199	0
80	11:57:18 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	74	Success	232	73	0
81	11:57:26 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	103	Success	232	101	0
82	11:57:26 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	66	Success	232	65	0
83	11:57:27 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	184	Success	232	183	0
84	11:57:27 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	127	Success	232	126	0
85	11:57:27 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	92	Success	232	90	0
86	11:57:27 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	111	Success	232	109	0
87	11:57:27 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	136	Success	232	135	0
88	11:57:27 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	137	Success	232	136	0
89	11:57:27 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	153	Success	232	152	0
90	11:57:27 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	226	Success	232	225	0
91	11:57:35 PM	Grupo de Usuarios 1-6	/SICA/registrarPagosRealizados.htm	168	Success	232	166	0
92	11:57:35 PM	Grupo de Usuarios 1-8	/SICA/registrarPagosRealizados.htm	59	Success	232	57	0
93	11:57:35 PM	Grupo de Usuarios 1-5	/SICA/registrarPagosRealizados.htm	102	Success	232	101	0
94	11:57:35 PM	Grupo de Usuarios 1-7	/SICA/registrarPagosRealizados.htm	62	Success	232	61	0
95	11:57:36 PM	Grupo de Usuarios 1-2	/SICA/registrarPagosRealizados.htm	60	Success	232	59	0
96	11:57:36 PM	Grupo de Usuarios 1-3	/SICA/registrarPagosRealizados.htm	54	Success	232	53	0
97	11:57:36 PM	Grupo de Usuarios 1-10	/SICA/registrarPagosRealizados.htm	78	Success	232	77	0
98	11:57:36 PM	Grupo de Usuarios 1-9	/SICA/registrarPagosRealizados.htm	160	Success	232	159	0
99	11:57:36 PM	Grupo de Usuarios 1-4	/SICA/registrarPagosRealizados.htm	121	Success	232	121	0
100	11:57:36 PM	Grupo de Usuarios 1-1	/SICA/registrarPagosRealizados.htm	78	Success	232	76	0

Tabla 52. Resultados en Tabla para el Proceso de Pagos

Fuente: Elaborado por el Autor

6.1.3.2. REPORTE RESUMEN

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/SICA/registrarPagosRealizados.htm	100	152	47	426	79.47	0.00%	1.3	0.29	232
Total	100	152	47	426	79.47	0.00%	1.3	0.29	232

Tabla 53. Reporte Resumen para el Proceso de Pagos

Fuente: Elaborado por el Autor

6.1.4. INDICADOR 01: TIEMPO DE GENERACIÓN EN REPORTES

Los resultados de este indicador nos demuestran que el tiempo de generación de reportes utilizando SICA es alrededor de 1 seg reduciendo enormemente las horas hombre que se puede utilizar para la obtención de información. **(ANEXO 18)**

6.1.5. INDICADOR 02: NÚMERO DE ÓRDENES DE PAGOS DUPLICADAS

Los resultados obtenidos luego de aplicar la guía de observación nro. 2 se aprecia que al no contar con un sistema se tiende a producir errores operativos, en este caso que se han realizado pagos duplicados y que aún se tengan deudas pendientes. El SICA resuelve este problema debido que realiza validaciones internas y automáticas que permitirá que no ocurra este problema. **(ANEXO 19)**

6.1.6. INDICADOR 03: GRADO DE SATISFACCIÓN

Para medir este indicador se realizó (02) encuestas una para usuarios del nivel estratégico (**ANEXO 21**) y la otra pa usuarios del nivel operacional (**0**). Cabe mencionar que mi población sólo es de (05) personas por lo tanto no se aplica el uso de los métodos estadísticos.

Los resultados obtenidos de la Encuesta Nro. 1 se puede inferir que el 100% de los usuarios de Nivel Operación está satisfecho con el SICA.

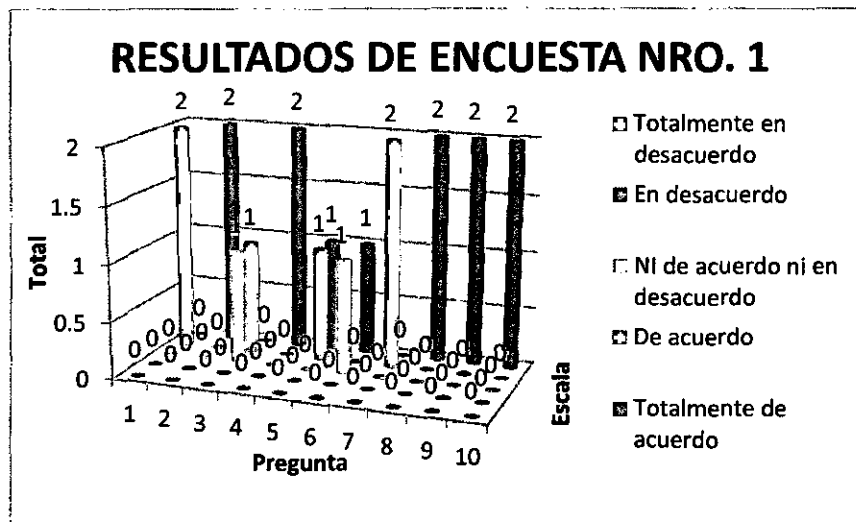


Figura 63. Gráfico de resultados para la Encuesta Nro. 1
Fuente: Elaborado por el Autor

Los resultados obtenidos de la Encuesta Nro. 2 se puede inferir que el 66.6667% de los usuarios del Nivel Estratégico está satisfecho con el SICA.

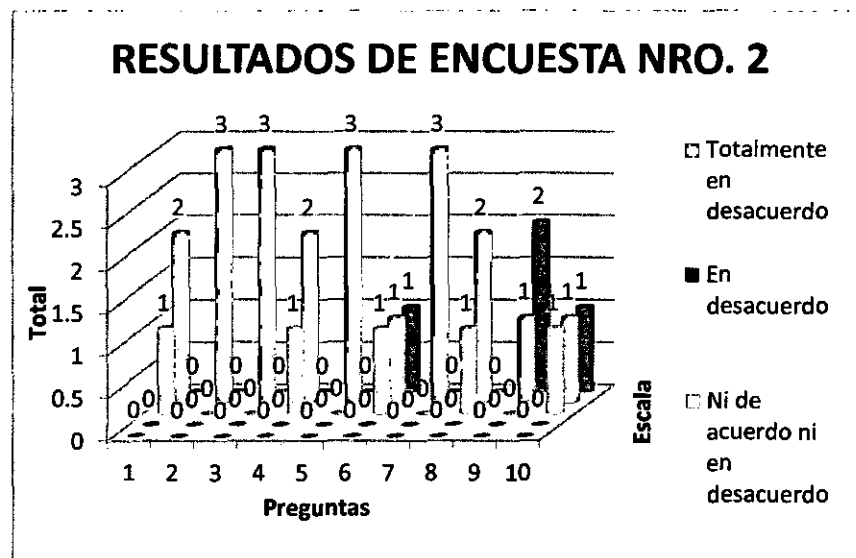


Figura 64. . Gráfico de resultados para la Encuesta Nro. 2
Fuente: Elaborado por el Autor

6.2. ANÁLISIS COSTO – BENEFICIO DEL SISTEMA

El análisis de costo – beneficio se utiliza para evaluar si SICA mediante la enumeración y valoración posterior en términos monetarios de todos los costos y beneficios derivados directa e indirectamente. Los beneficios y costos se pueden representar como tangibles o intangibles.

6.2.1. BENEFICIOS TANGIBLES

Son ventajas que se pueden medir en soles que se acreditan a la EPS Grau mediante el uso del SICA.

- Aumento de velocidad de procesamiento.
- Acceso a la información de las acreencias en una forma más oportuna.
- Emisión de pagos de manera más rápida y sin generar pagos dobles.

6.2.2. BENEFICIOS INTAGIBLES

Son aquellas que no puede ser cuantificadas pero que no dejan de ser relavantes para la EPS Grau.

- Mejoramiento en el proceso de tomas de decisiones
- Mayor control de los documentos que genera el proceso concursal de la EPS Grau.
- Mejor imagen y reputación corporativa de la EPS Grau.

6.2.3. COMPARACION CON EL SICA

En esta sección se presentará un cuadro comparativo (Sin el SICA y con el SICA) de los costos de los procesos más importantes que es la obtención de información de una acreencia y la emisión de pagos sin correr el riesgo de error operativo de realizar pagos duplicados.

SIN SICA					
CONCEPTO	CANTIDAD PERSONAS	TIEMPO	SUELDO (S/.)	COSTO TOTAL (S/.)	OBSERVACIONES
Acceso de la información de una Acreencia	1	1 Semana	2000.00	333.3335	<ul style="list-style-type: none">• Se trabaja de lunes a viernes, 8 horas diarias.• Costo x día: S/. 66.6667• Costro x hora: S/. 8.3333

					• Costo x Segundo: S/. 0.0023.
Emisión de pagos	1	3 días	2000.00	200.0001	Se genera 600 pagos.
			TOTAL	533.3336	
CON SICA					
CONCEPTO	CANTIDAD PERSONAS	TIEMPO	SUELDO (S/.)	COSTO TOTAL (S/.)	OBSERVACIONES
Acceso de la información de una Acreencia	1	3 seg.	2000.00	0.0069	El tiempo restante se puede dedicar atender otras actividades. En 600 pagos será 38 709 seg
Emisión de pagos	1	64.515 seg.	2000.00	83.3333	equivale a 10.7525 horas = 1 día + 2.7525 hora
			TOTAL	83.3402	

Tabla 54. Cuadro comparativo de Costos
Fuente: Elaborado por el Autor

A partir del cuadro anterior se puede concluir que definitivamente la implementación del SICA reduce considerablemente los costos lo cual permite que sea un proyecto viable.

CONCLUSIONES

Luego de terminar con el proyecto SICA y en base a los resultados obtenidos se concluye lo siguiente:

- La automatización del proceso concursal de la EPS Grau mediante el SICA permitirá reducir costos, generar valor agregado y gestionar de una manera óptima las acreencias, deudas y pagos que se realizan.
- La reducción en la búsqueda de acreencias, deudas, etc y generación de reportes de los mismo es notable reduciendo las horas hombres permitiendo que el tiempo que se gana se puedan realizar otras actividades importantes.
- Utilizar y aplicar Scrum permitió que no sólo se obtenga un prototipo sino que el entregable final este funcionando en un 100%, para lo cual se debe cumplir con los cronogramas establecidos en los sprints.
- Utilizar Spring permite que SICA tenga una arquitectura robusta, segura, escalable y de fácil mantenimiento.
- Elaborar la base de datos en MySQL permite albergar la data de SICA en la cual se observa que las tablas principales donde gira toda la lógica del negocio son la de acreencia, deuda concursal, programación de pago y pagos.
- Los resultados de los indicadores fueron satisfactorio demostrando que SICA es eficiente y que los usuarios están satisfecho con el SICA.
- La aplicación adecuada de las TIC mejora enormemente el proceso en el cual se piensa aplicar, como es en el caso de este proyecto de investigación.

RECOMENDACIONES

Al final de desarrollar SICA y con la experiencia obtenida se recomienda lo siguiente:

- Migrar la base de datos a PostgreSQL debido que este se acerca en cuanto a funcionalidad y robustez a Oracle. Teniendo un lenguaje y estructura y administración parecida, además que la EPS Grau la mayoría de sus sistemas usan PostgreSQL.
- Realizar los diseños de los reportes con BirtReport debido que es más flexible, es más rápido que los JasperReports, mantenido por una comunidad, interfaz más amigable e intuitiva.
- Se recomienda manejar estándares en la programación para que el código sea como si un mismo programador lo ha realizado.
- Documentar el código como en los objetos de base permite que cualquier programador le permita conocer y familiarizarse con el SICA y esto con lo mencionado en el punto anterior permite que el mantenimiento de SICA se realice de una mejor manera.
- Utilizar un sistema de control de versiones como Subversion o GitHub en un servidor de desarrollo para que ahí se conserve la versión final del SICA y para llevar un registro de los cambios realizados en el proyecto.

BIBLIOGRAFÍA

- Alvarez Caules, C. (2012). *Arquitectura Java Sólida*.
- Anacleto, V. A. (2014, Octubre 07). *Epidata Consulting*. Retrieved from <http://www.epidataconsulting.com/adjuntos/UML%202.0%20-%20CODE.pdf>
- Beck, K. (1999). *Extreme Programming Explained*. EMBRACE CHANGE.
- Blanck, P. (2008). *Eficiencia y Racionalidad en el Sistema Concursal. El Caso Peruano*.
- Cohen Karen, D., & Asín Lares, E. (2000). *Sistemas de Información para los Negocios*. México: McGraw-Hill/Interamericana Editores S.A.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *Una Introducción básica a la teoría y práctica de Scrum*. Projectalis.
- Dessi, M. (2009). *Spring 2.5 Aspect - Oriented Programming*. UK: Packt Publishing Ltd.
- Flores, C. A. (2013, Noviembre). *Análisis, Diseño e Implementación de un Sistema de Recaudación de Deudas*. Lima, Perú.
- Hamilton, K., & Miles, R. (2006). *Learning UML 2.0*. O'Reilly.
- Kendall, K. E., & Kendall, J. E. (2005). *Análisis y Diseño de Distemas, Sexta edición*. México: Pearson Educacion.
- Kniberg, H. (2007). *Scrum y XP desde las trincheras*. Estados Unidos de América: C4Media.
- Microsoft. (n.d.). *Developer Network*. Retrieved Mayo 10, 2014, from <http://msdn.microsoft.com/es-es/library/bb972214.aspx#XSLTsection122121120120>
- Pérez, S. A. (2009, Mayo). *Diseño e Implementación de una Plataforma de Telecombranzas Integrado al Sistema E-Government de una Empresa de Recaudación Tributaria*. Lima, Perú.
- Perú, P. U. (n.d.). Retrieved Julio 04, 2014, from *Proyectos de Fin de Carrera de Ingeniería Informática*: <http://inform.pucp.edu.pe/~inf391/>
- Quito Rodríguez, C. Z., & León García, T. (2013, Octubre). *Módulo Guía de Investigación*. Piura, Perú: Facultad de Ingeniería Industrial - UNP.
- República, C. d. (2002, Agosto). *Ley General del Sistema Concursal. Ley N° 278091*. Perú.
- Schmuller, J. (2003). *Aprendiendo UML en 24 Horas*. México: Pearson Educación Latinoamérica.
- Sommerville, I. (2005). *Ingeniería del Software Séptima edición*. Madrid: Pearson.

Thomas Risberg, R. E. (n.d.). *Spring Framework*. Retrieved Mayo 7, 2014, from <http://docs.spring.io/docs/Spring-MVC-step-by-step/>

Walls, C. (2011). *Spring in Action Third Edition*. United States of America: Manning.

Winch, R., & Mularien, P. (2012). *Spring Security 3.1*. UK: Packt Publishing.

ANEXOS

ANEXO 1. DEUDA DE CARÁCTER CONCURSAL O ESTRUCTURAL DE DIFERENTE NATURALEZA

Tipo de Crédito	Total Reconocido	Total No Reconocido	Contingente	Total General
Comercial	194,355,264.82	3,698,989.63	104,973,814.68	303,028,051.13
Tributario	50,464,018.91	336,126.28		50,800,145.19
Laboral	20,924,698.44	755,179.94	997,983.38	22,677,861.76
Sub total Previsional	6,459,058.71		192,500.03	6,651,657.74
Sub Total Proc Laborales	1,502,854.13	420,911.99	478,090.20	2,401,856.32
Sub Total Beneficio Sociales	1,968,542.55	165,449.63		2,133,992.18
Sub Total CTS	10,790,159.06	85,812.32		10,875,971.39
Sub Total Colaterales	141,394.03	37,568.07		178,962.10
Sub Total Otros Conceptos Laborales	33,807.47		327,294.15	361,101.62
Sub Total Asignación Familiar	2,298.00			2,298.00
Sub Total Pensión de viudez	26,584.48			26,584.48
Sub Total Retenciones		45,437.93		45,437.93
Otros	4,485.00			4,485.00
Quirografirios	12,870.60			12,870.60
Total	265,761,319.77	4,790,295.85	105,971,798.06	376,523,413.69

ANEXO 2. registrarAcreencia.jsp

```
<%--
  Document : registrarAcreencia
  Created on : Apr 10, 2015, 9:44:28 PM
  Author : Carnol
--%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <jsp:include page="../templates/_header.jsp"/>
    <script src="<%=request.getContextPath()%>/BASE-INF/js/registrar/registrarAcreencia.js"></script>
  </head>
  <body>
    <jsp:include page="../templates/_navbar.jsp"/>
    <jsp:include page="../templates/_credits.jsp"/>
    <div class="container">
      <div class="row">
        <div class="col-md-10 col-md-offset-1">
          <div class="panel panel-primary">
            <div class="panel-heading">Nueva Acreencia</div>
            <div class="panel-body">
              <form id="formAcreencia" class="form-horizontal" action="#"
method="POST">
                <input type="hidden" class="form-control" id="hddUserAdd" />
                <input type="hidden" class="form-control" id="hddDateAdd" />
                <div class="form-group">
                  <label for="inputEmail3" class="col-md-4 control-label">Nro.
Acreencia</label>
                  <div class="col-md-4">
                    <input type="text" class="form-control" id="txtNroAcreencia"
readonly>
                  </div>
                </div>
                <div class="form-group">
                  <label for="inputEmail3" class="col-md-4 control-label">Nro.
Deuda Concursal (*)</label>
                  <div class="col-md-3">
                    <input type="text" class="form-control"
id="txtNroDeudaConcursal">
                  </div>
                <button id="btnFind" type="submit" class="btn btn-primary">
                  <span class="glyphicon glyphicon-search" aria-
hidden="true"></span>
                  Buscar
                </button>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

<div id="msj-info" class="alert alert-info">
  <strong>MENSAJE:</strong>
  No se encontraron resultados para este Numero de Deuda Concursal.
</div>
<div class="form-group">
  <label for="inputEmail3" class="col-md-4 control-label">Acreedor</label>
  <input type="hidden" class="form-control" id="hddAcreedor" />
  <div class="col-md-6">
    <input type="text" class="form-control" id="txtAcreedor">
  </div>
</div>
<div class="form-group">
  <label for="inputEmail3" class="col-md-4 control-label">Tipo
Acreedor</label>
  <div class="col-md-4">
    <input type="text" class="form-control" id="txtTipoAcreedor"
readonly>
  </div>
</div>
<div class="form-group">
  <label for="inputEmail3" class="col-md-4 control-label">Tipo
Deuda</label>
  <div class="col-md-4">
    <input type="text" class="form-control" id="txtTipDeu"
readonly>
    <input type="hidden" class="form-control" id="hddIdTipDeu"
readonly>
    <input type="hidden" class="form-control" id="idTipoAcreencia"
readonly>
  </div>
</div>
<div class="form-group" id="tipo_acreencia">
  <label for="inputEmail3" class="col-md-4 control-label">Tipo
Acreencia</label>
  <div id="div_tip_acreencia" class="col-md-4">
  </div>
</div>
<div class="form-group">
  <label for="inputEmail3" class="col-md-4 control-label">Nro.
Documento (*)</label>
  <div class="col-md-6">
    <input type="text" class="form-control" id="txtNroDocumento" >
  </div>
</div>
<div class="form-group">
  <label for="inputEmail3" class="col-md-4 control-label">Monto
Reconocido (*)</label>
  <div class="col-md-6">

```

```

        <input type="text" class="form-control"
id="txtMontoReconocido">
    </div>
</div>
<div class="form-group">
    <label for="inputEmail3" class="col-md-4 control-
label">Resoluci&oacute;n (*)</label>
    <div class="col-md-6">
        <input type="text" class="form-control" id="txtResolucion">
    </div>
</div>
<div class="form-group">
    <label for="inputEmail3" class="col-md-4 control-label">Fecha
Resoluci&oacute;n (*)</label>
    <div class='col-md-4 input-group date' id="fecha_resolucion">
        <input type='text' class="form-control" id="txtFechaResolucion"
readonly />
        <span class="input-group-addon">
            <span class="glyphicon glyphicon-calendar"></span>
        </span>
    </div>
</div>
<div class="form-group">
    <label for="inputEmail3" class="col-md-4 control-label">Fecha
Registro (*)</label>
    <div class='col-md-4 input-group date' id="fecha_registro">
        <input type='text' class="form-control" id="txtFechaRegistro"
readonly />
        <span class="input-group-addon">
            <span class="glyphicon glyphicon-calendar"></span>
        </span>
    </div>
</div>
<div class="form-group">
    <label for="inputEmail3" class="col-md-4 control-label">Indicador
Activo</label>
    <div class="col-md-4">
        <input id="ckbIndAct" type="checkbox" checked>
    </div>
</div>
<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        <button id="btnNew" type="submit" class="btn btn-primary">
            <span class="glyphicon glyphicon-plus" aria-
hidden="true"></span>
            Nuevo
        </button>
        <button id="btnAdd" type="submit" class="btn btn-primary">
            <span class="glyphicon glyphicon-floppy-saved" aria-
hidden="true"></span>

```



```

        Registrar
    </button>
    <button id="btnUpdate" type="submit" class="btn btn-primary">
        <span class="glyphicon glyphicon-edit" aria-
hidden="true"></span>
        Modificar
    </button>
    <button id="btnCancel" type="submit" class="btn btn-primary">
        <span class="glyphicon glyphicon-repeat" aria-
hidden="true"></span>
        Cancelar
    </button>
    <button id="btnList" type="submit" class="btn btn-primary">
        <span class="glyphicon glyphicon-th-list" aria-
hidden="true"></span>
        Listar
    </button>
</div>
</div>
<div class="alert alert-info">
    <strong>(*) Campos obligatorios.</strong>
</div>
<div id="msjResultados">
    <div id="msj-ok" class="alert alert-success">
        <strong>EXITO:</strong>
        Proceso ejecutado correctamente
    </div>
    <div id="msj-error" class="alert alert-danger">
        <strong>VERIFICAR:</strong>
        Se ha producido un error.
    </div>
</div>
</div>
</form>
</div>
</div>
<div id="tablaResultados">
</div>
</div>
</div>
</body>
</html>

```

ANEXO 3. ENTIDAD ACREEDOR (Acreedor.java)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.entity;

import java.io.Serializable;
import java.util.Date;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;
import org.codehaus.jackson.annotate.JsonIgnore;

/**
 *
 * @author Carnol
 */
@Entity
@Table(name = "acreedor", catalog = "bdsica", schema = "")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Acreedor.findAll", query = "SELECT a FROM Acreedor a"),
    @NamedQuery(name = "Acreedor.findByIdAcreedor", query = "SELECT a FROM Acreedor a WHERE a.idAcreedor = :idAcreedor"),
    @NamedQuery(name = "Acreedor.findByUserAdd", query = "SELECT a FROM Acreedor a WHERE a.userAdd = :userAdd"),
    @NamedQuery(name = "Acreedor.findByDateAdd", query = "SELECT a FROM Acreedor a WHERE a.dateAdd = :dateAdd"),
    @NamedQuery(name = "Acreedor.findByUserUpdate", query = "SELECT a FROM Acreedor a WHERE a.userUpdate = :userUpdate"),
    @NamedQuery(name = "Acreedor.findByDateUpdate", query = "SELECT a FROM Acreedor a WHERE a.dateUpdate = :dateUpdate"),
```

```

    @NamedQuery(name = "Acreeedor.findByIndActivo", query = "SELECT a FROM
Acreeedor a WHERE a.indActivo = :indActivo"),
    @NamedQuery(name = "Acreeedor.findByTipoAcreeedor", query = "SELECT a FROM
Acreeedor a WHERE a.tipoAcreeedor = :tipoAcreeedor"))
public class Acreeedor implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idAcreeedor", nullable = false)
    private Integer idAcreeedor;
    @Size(max = 45)
    @Column(name = "userAdd", length = 45)
    private String userAdd;
    @Column(name = "dateAdd")
    @Temporal(TemporalType.TIMESTAMP)
    private Date dateAdd;
    @Size(max = 45)
    @Column(name = "userUpdate", length = 45)
    private String userUpdate;
    @Column(name = "dateUpdate")
    @Temporal(TemporalType.TIMESTAMP)
    private Date dateUpdate;
    @Size(max = 1)
    @Column(name = "indActivo", length = 1)
    private String indActivo;
    @Column(name = "tipoAcreeedor")
    private Integer tipoAcreeedor;
    @OneToMany(mappedBy = "idAcreeedor", fetch = FetchType.LAZY)
    private List<Deudaconcurasal> deudaconcurasalList;
    @OneToMany(mappedBy = "idAcreeedor", fetch = FetchType.LAZY)
    private List<Pago> pagoList;
    @OneToMany(mappedBy = "idAcreeedor", fetch = FetchType.LAZY)
    private List<Trabajador> trabajadorList;
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "acreeedor")
    private List<BeneficiarioAcreeedor> beneficiarioAcreeedorList;
    @OneToMany(mappedBy = "idAcreeedor", fetch = FetchType.LAZY)
    private List<Proveedor> proveedorList;

    public Acreeedor() {
    }

    public Acreeedor(Integer idAcreeedor) {
        this.idAcreeedor = idAcreeedor;
    }

    public Integer getIdAcreeedor() {
        return idAcreeedor;
    }
}

```

```

public void setIdAcreeador(Integer idAcreeador) {
    this.idAcreeador = idAcreeador;
}

public String getUserAdd() {
    return userAdd;
}

public void setUserAdd(String userAdd) {
    this.userAdd = userAdd;
}

public Date getDateAdd() {
    return dateAdd;
}

public void setDateAdd(Date dateAdd) {
    this.dateAdd = dateAdd;
}

public String getUserUpdate() {
    return userUpdate;
}

public void setUserUpdate(String userUpdate) {
    this.userUpdate = userUpdate;
}

public Date getDateUpdate() {
    return dateUpdate;
}

public void setDateUpdate(Date dateUpdate) {
    this.dateUpdate = dateUpdate;
}

public String getIndActivo() {
    return indActivo;
}

public void setIndActivo(String indActivo) {
    this.indActivo = indActivo;
}

public Integer getTipoAcreeador() {
    return tipoAcreeador;
}

public void setTipoAcreeador(Integer tipoAcreeador) {
    this.tipoAcreeador = tipoAcreeador;
}

```

```

    }

    @XmlTransient
    @JsonIgnore
    public List<Deudaconcurasal> getDeudaconcurasalList() {
        return deudaconcurasalList;
    }

    public void setDeudaconcurasalList(List<Deudaconcurasal> deudaconcurasalList) {
        this.deudaconcurasalList = deudaconcurasalList;
    }

    @XmlTransient
    @JsonIgnore
    public List<Pago> getPagoList() {
        return pagoList;
    }

    public void setPagoList(List<Pago> pagoList) {
        this.pagoList = pagoList;
    }

    @XmlTransient
    @JsonIgnore
    public List<Trabajador> getTrabajadorList() {
        return trabajadorList;
    }

    public void setTrabajadorList(List<Trabajador> trabajadorList) {
        this.trabajadorList = trabajadorList;
    }

    @XmlTransient
    @JsonIgnore
    public List<BeneficiarioAcreedor> getBeneficiarioAcreedorList() {
        return beneficiarioAcreedorList;
    }

    public void setBeneficiarioAcreedorList(List<BeneficiarioAcreedor>
beneficiarioAcreedorList) {
        this.beneficiarioAcreedorList = beneficiarioAcreedorList;
    }

    @XmlTransient
    @JsonIgnore
    public List<Proveedor> getProveedorList() {
        return proveedorList;
    }

    public void setProveedorList(List<Proveedor> proveedorList) {

```

```

        this.proveedorList = proveedorList;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idAcreedor != null ? idAcreedor.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Acreedor)) {
            return false;
        }
        Acreedor other = (Acreedor) object;
        if (((this.idAcreedor == null && other.idAcreedor != null) || (this.idAcreedor != null
        && !this.idAcreedor.equals(other.idAcreedor))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.epsgrau.canm.entity.Acreedor[ idAcreedor=" + idAcreedor + " ]";
    }
}

```

ANEXO 4. AcreedorDAO.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.dao;

import com.epsgrau.canm.dto.AutoCompleteDTO;
import com.epsgrau.canm.entity.Acreedor;
import com.epsgrau.canm.helpers.ExcepcionBusiness;
import com.epsgrau.canm.helpers.HelperHibernate;
import com.epsgrau.canm.helpers.HelperMensajes;
import com.epsgrau.canm.dto.DatoDTO;
import com.epsgrau.canm.dto.TrabajadorProveedorDTO;
import java.util.List;
import org.hibernate.Hibernate;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

```

```

import org.hibernate.transform.Transformers;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataAccessException;
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
import org.springframework.stereotype.Repository;

/**
 *
 * @author Carnol
 */
@Repository
public class AcreedorDAO extends HibernateDaoSupport {

    private final HelperMensajes mensajes;
    private static org.slf4j.Logger loggerBD =
        LoggerFactory.getLogger(AcreedorDAO.class);

    @Autowired
    public AcreedorDAO(SessionFactory sf, HelperMensajes mensajes) {
        super.setSessionFactory(sf);
        this.mensajes = mensajes;
    }

    public void guardarActualizar(Acreedor acreedor) throws Exception {
        try {
            getHibernateTemplate().saveOrUpdate(acreedor);
            loggerBD.info("guardarActualizar: Proceso se ejecuto correctamente.");
        } catch (Exception e) {
            loggerBD.error(e.getMessage() + "\tCausa: " + e.getCause());
            throw new ExcepcionBusiness(mensajes.getErrorOtherException());
        }
    }

    public Acreedor buscarPorId(int id) throws DataAccessException {
        return getHibernateTemplate().get(Acreedor.class, id);
    }

    public List<Acreedor> listar() throws DataAccessException {
        List<Acreedor> lista =
            getHibernateTemplate().findByNameQuery("Acreedor.findAll");
        return lista;
    }

    public int obtenerUltimoID() throws Exception {
        DatoDTO acreedor = null;
        Session ses =
            HelperHibernate.getSessionHibernateTemplate(getHibernateTemplate());
        acreedor = (DatoDTO) ses.createSQLQuery("CALL re_p_getLastIdAcreedor()")
            .addScalar("ultimoID", Hibernate.INTEGER)

```

```

.setResultTransformer(Transformers.aliasToBean(DatoDTO.class)).uniqueResult();
    return acreedor.getUltimoID();
}

public void eliminar(Acreedor acreed) throws DataAccessException, ExcepcionBusiness
{
    try {
        getHibernateTemplate().delete(acreed);
        loggerBD.info("eliminar: Proceso se ejecuto correctamente.");
    } catch (Exception e) {
        loggerBD.error(e.getMessage() + "\tCausa: " + e.getCause());
        throw new ExcepcionBusiness(mensajes.getErrorOtherException());
    }
}

public List<TrabajadorProveedorDTO> listarTrabajadorProveedor(String buscar) {
    List<TrabajadorProveedorDTO> lista = null;
    Session ses =
    HelperHibernate.getSessionHibernateTemplate(getHibernateTemplate());
    lista = ses.createQuery("CALL re_p_listar_trab_prov(:buscar)")
        .addScalar("idAcreedor", Hibernate.INTEGER)
        .addScalar("documento", Hibernate.STRING)
        .addScalar("nombre1", Hibernate.STRING)
        .addScalar("nombre2", Hibernate.STRING)
        .addScalar("tipo", Hibernate.STRING)
        .setString("buscar", buscar)

.setResultTransformer(Transformers.aliasToBean(TrabajadorProveedorDTO.class)).list();
    return lista;
}

public List<AutocompleteDTO> listarTrabajadorAutocomplete(String buscar) {
    List<AutocompleteDTO> lista = null;
    Session ses =
    HelperHibernate.getSessionHibernateTemplate(getHibernateTemplate());
    lista = ses.createQuery("CALL rep_p_auto_trab_prov(:rpOpcion, :buscar)")
        .addScalar("id", Hibernate.INTEGER)
        .addScalar("fullname", Hibernate.STRING)
        .setString("rpOpcion", "1")
        .setString("buscar", buscar)

.setResultTransformer(Transformers.aliasToBean(AutocompleteDTO.class)).list();
    return lista;
}

public List<AutocompleteDTO> listarProveedorAutocomplete(String buscar) {
    List<AutocompleteDTO> lista = null;
    Session ses =
    HelperHibernate.getSessionHibernateTemplate(getHibernateTemplate());

```



```

        lista = ses.createQuery("CALL rep_p_auto_trab_prov(:rpOpcion, :buscar)")
            .addScalar("id", Hibernate.INTEGER)
            .addScalar("fullname", Hibernate.STRING)
            .setString("rpOpcion", "2")
            .setString("buscar", buscar)

        .setResultTransformer(Transformers.aliasToBean(AutocompleteDTO.class)).list();
        return lista;
    }

    public List<AutocompleteDTO> listarAcreeedorAutocomplete(String buscar) {
        List<AutocompleteDTO> lista = null;
        Session ses =
        HelperHibernate.getSessionHibernateTemplate(getHibernateTemplate());
        lista = ses.createQuery("CALL rep_p_auto_acreeedor(:buscar)")
            .addScalar("id", Hibernate.INTEGER)
            .addScalar("fullname", Hibernate.STRING)
            .setString("buscar", buscar)

        .setResultTransformer(Transformers.aliasToBean(AutocompleteDTO.class)).list();
        return lista;
    }
}

```

ANEXO 5. AcreeedorBusiness.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.business;

import com.epsgrau.canm.dao.AcreeedorDAO;
import com.epsgrau.canm.dto.AutocompleteDTO;
import com.epsgrau.canm.entity.Acreeedor;
import com.epsgrau.canm.helpers.ExcepcionBusiness;
import com.epsgrau.canm.dto.TrabajadorProveedorDTO;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataAccessException;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author Carnol
 */
@Component
public class AcreeedorBusiness {

```

```

@Autowired
private AcreedorDAO acreedorDAO;

@Transactional(rollbackFor = {Exception.class})
public void guardarActualizar(Acreedor acreedor) throws Exception {
    acreedorDAO.guardarActualizar(acreedor);
}

public Acreedor buscarPorId(int id) throws DataAccessException {
    return acreedorDAO.buscarPorId(id);
}

public List<Acreedor> listar() throws DataAccessException {
    return acreedorDAO.listar();
}

@Transactional(rollbackFor = {Exception.class})
public int obtenerUltimoID() throws Exception {
    return acreedorDAO.obtenerUltimoID();
}

@Transactional(rollbackFor = {Exception.class})
public void eliminar(Acreedor acreed) throws DataAccessException, ExcepcionBusiness
{
    acreedorDAO.eliminar(acreed);
}

@Transactional(rollbackFor = {Exception.class})
public List<TrabajadorProveedorDTO> listarTrabajadorProveedor(String buscar) {
    return acreedorDAO.listarTrabajadorProveedor(buscar);
}

@Transactional(rollbackFor = {Exception.class})
public List<AutocompleteDTO> listarTrabajadorAutocomplete(String buscar) {
    return acreedorDAO.listarTrabajadorAutocomplete(buscar);
}

@Transactional(rollbackFor = {Exception.class})
public List<AutocompleteDTO> listarProveedorAutocomplete(String buscar) {
    return acreedorDAO.listarProveedorAutocomplete(buscar);
}

@Transactional(rollbackFor = {Exception.class})
public List<AutocompleteDTO> listarAcreedorAutocomplete(String buscar) {
    return acreedorDAO.listarAcreedorAutocomplete(buscar);
}
}

```

ANEXO 6. AcreeedorService.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.service;

import com.epsgrau.canm.business.AcreeedorBusiness;
import com.epsgrau.canm.dto.AutoCompleteDTO;
import com.epsgrau.canm.entity.Acreeedor;
import com.epsgrau.canm.helpers.ExcepcionBusiness;
import com.epsgrau.canm.dto.TrabajadorProveedorDTO;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataAccessErrorException;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

/**
 *
 * @author Carnol
 */
@Service
public class AcreeedorService {

    @Autowired
    private AcreeedorBusiness acreeedorBusiness;

    @Transactional(rollbackFor = {Exception.class})
    public void guardarActualizar(Acreeedor acreeedor) throws Exception {
        acreeedorBusiness.guardarActualizar(acreeedor);
    }

    public Acreeedor buscarPorId(int id) throws DataAccessErrorException {
        return acreeedorBusiness.buscarPorId(id);
    }

    public List<Acreeedor> listar() throws DataAccessErrorException {
        return acreeedorBusiness.listar();
    }

    @Transactional(rollbackFor = {Exception.class})
    public int obtenerUltimoID() throws Exception {
        return acreeedorBusiness.obtenerUltimoID();
    }

    @Transactional(rollbackFor = {Exception.class})
    public void eliminar(Acreeedor acreeedor) throws DataAccessErrorException, ExcepcionBusiness
    {
```

```

        acreedorBusiness.eliminar(acreed);
    }

    @Transactional(rollbackFor = {Exception.class})
    public List<TrabajadorProveedorDTO> listarTrabajadorProveedor(String buscar) {
        return acreedorBusiness.listarTrabajadorProveedor(buscar);
    }

    @Transactional(rollbackFor = {Exception.class})
    public List<AutocompleteDTO> listarTrabajadorAutocomplete(String buscar) {
        return acreedorBusiness.listarTrabajadorAutocomplete(buscar);
    }

    @Transactional(rollbackFor = {Exception.class})
    public List<AutocompleteDTO> listarProveedorAutocomplete(String buscar) {
        return acreedorBusiness.listarProveedorAutocomplete(buscar);
    }

    @Transactional(rollbackFor = {Exception.class})
    public List<AutocompleteDTO> listarAcreedorAutocomplete(String buscar) {
        return acreedorBusiness.listarAcreedorAutocomplete(buscar);
    }
}

```

ANEXO 7. AcreedorController.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.controller;

import com.epsgrau.canm.dto.AutocompleteDTO;
import com.epsgrau.canm.dto.MensajeDTO;
import com.epsgrau.canm.helpers.HelperMensajes;
import com.epsgrau.canm.service.AcreedorService;
import java.util.ArrayList;
import java.util.List;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

/**
 *
 * @author Carnol
 */

```

```

*/
@Controller
public class AcreedorController {

    @Autowired
    private AcreedorService as;

    @Autowired
    private HelperMensajes mensajes;

    @Autowired
    private MensajeDTO mensaje;

    private static org.slf4j.Logger loggerController =
    LoggerFactory.getLogger(AcreedorController.class);

    @RequestMapping(value = "listarTrabajadorAutocomplete.htm", method =
    RequestMethod.GET)
    public @ResponseBody
    List<AutocompleteDTO> listarTrabajadorAutocomplete(@RequestParam String
    tagName) {
        try {
            loggerController.info("listarTrabajadorAutocomplete: Proceso ejecutado
            correctamente");
            return as.listarTrabajadorAutocomplete(tagName);
        } catch (Exception e) {
            loggerController.error(e.getMessage() + "\tCausa: " + e.getCause());
            return new ArrayList<AutocompleteDTO>();
        }
    }

    @RequestMapping(value = "listarProveedorAutocomplete.htm", method =
    RequestMethod.GET)
    public @ResponseBody
    List<AutocompleteDTO> listarProveedorAutocomplete(@RequestParam String
    tagName) {
        try {
            loggerController.info("listarProveedorAutocomplete: Proceso ejecutado
            correctamente");
            return as.listarProveedorAutocomplete(tagName);
        } catch (Exception e) {
            loggerController.error(e.getMessage() + "\tCausa: " + e.getCause());
            return new ArrayList<AutocompleteDTO>();
        }
    }

    @RequestMapping(value = "listarAcreedorAutocomplete.htm", method =
    RequestMethod.GET)
    public @ResponseBody

```

```

    List<AutocompleteDTO> listarAcreeadorAutocomplete(@RequestParam String
tagName) {
    try {
        loggerController.info("listarAcreeadorAutocomplete: Proceso ejecutado
correctamente");
        return as.listarAcreeadorAutocomplete(tagName);
    } catch (Exception e) {
        loggerController.error(e.getMessage() + "\tCausa: " + e.getCause());
        return new ArrayList<AutocompleteDTO>();
    }
}
}

```

ANEXO 8. AcreeadorDTO.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.dto;

import java.io.Serializable;

/**
 *
 * @author Camol
 */
public class AcreeadorDTO implements Serializable {

    private Integer idAcreeador;
    private String acreeador;
    private String tipoAcreeador;
    private Integer idTipoDeuda;
    private String tipDeuda;

    public AcreeadorDTO() {
    }

    public Integer getIdAcreeador() {
        return idAcreeador;
    }

    public void setIdAcreeador(Integer idAcreeador) {
        this.idAcreeador = idAcreeador;
    }

    public String getAcreeador() {
        return acreeador;
    }
}

```

```

public void setAcreeador(String acreeador) {
    this.acreeador = acreeador;
}

public String getTipoAcreeador() {
    return tipoAcreeador;
}

public void setTipoAcreeador(String tipoAcreeador) {
    this.tipoAcreeador = tipoAcreeador;
}

public Integer getIdTipoDeuda() {
    return idTipoDeuda;
}

public void setIdTipoDeuda(Integer idTipoDeuda) {
    this.idTipoDeuda = idTipoDeuda;
}

public String getTipDeuda() {
    return tipDeuda;
}

public void setTipDeuda(String tipDeuda) {
    this.tipDeuda = tipDeuda;
}
}

```

ANEXO 9. pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.epsgrau.canm</groupId>
    <artifactId>SICA</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>

    <name>SICA</name>

    <properties>
        <endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

```

```

    <version.spring>3.0.3.RELEASE</version.spring>
    <version.spring.aux>3.0.4.RELEASE</version.spring.aux>
    <version.spring.test>3.0.5.RELEASE</version.spring.test>
</properties>

<dependencies>
  <dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-web-api</artifactId>
    <version>7.0</version>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>org.eclipse.persistence</groupId>
    <artifactId>eclipselink</artifactId>
    <version>2.5.2</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.persistence</groupId>
    <artifactId>org.eclipse.persistence.jpa.modelgen.processor</artifactId>
    <version>2.5.2</version>
    <scope>provided</scope>
  </dependency>

  <!-- MySql -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <!--<version>5.1.37</version>-->
    <version>5.1.26</version>
  </dependency>

  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.4</version>
  </dependency>

  <!-- JSTL -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>

```



```

    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-asm</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-beans</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-expression</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>${version.spring.aux}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-orm</artifactId>
    <version>${version.spring.aux}</version>
</dependency>

<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-acl</artifactId>
    <version>3.0.3.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>

```

```

    <artifactId>spring-security-config</artifactId>
    <version>${version.spring}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
    <version>${version.spring}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-ldap</artifactId>
    <version>${version.spring}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-openid</artifactId>
    <version>${version.spring}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-taglibs</artifactId>
    <version>${version.spring}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>${version.spring}</version>
</dependency>

<!-- Test -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${version.spring.test}</version>
</dependency>

<!--conector postgresql-->

<!-- Transacciones -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-tx</artifactId>
    <version>${version.spring.aux}</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>${version.spring.aux}</version>
</dependency>

```

```

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${version.spring.aux}</version>
</dependency>

<!-- SQLite JDBC library -->
<dependency>
  <groupId>org.xerial</groupId>
  <artifactId>sqlite-jdbc</artifactId>
  <version>3.7.2</version>
</dependency>

<dependency>
  <groupId>taglibs</groupId>
  <artifactId>standard</artifactId>
  <version>1.1.2</version>
</dependency>

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate</artifactId>
  <version>3.2.5.ga</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-annotations</artifactId>
  <version>3.4.0.GA</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-commons-annotations</artifactId>
  <version>3.3.0.ga</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>3.3.2.GA</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>3.3.2.GA</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-c3p0</artifactId>
  <version>3.6.3.Final</version>
</dependency>

```

```

<!-- LOG4J -->
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.5.10</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.5.10</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>jcl-over-slf4j</artifactId>
  <version>1.5.10</version>
</dependency>

<dependency>
  <groupId>javax.transaction</groupId>
  <artifactId>jta</artifactId>
  <version>1.0.1B</version>
</dependency>

<dependency>
  <groupId>commons-logging</groupId>
  <artifactId>commons-logging</artifactId>
  <version>1.1.1</version>
</dependency>

<!-- Reportes -->
<dependency>
  <groupId>net.sf.jasperreports</groupId>
  <artifactId>jasperreports</artifactId>
  <version>5.1.0</version>
</dependency>
<dependency>
  <groupId>org.codehaus.groovy</groupId>
  <artifactId>groovy-all</artifactId>
  <version>2.0.1</version>
</dependency>
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>3.13</version>
</dependency>

<!-- JSON -->
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-core</artifactId>

```

```

    <version>2.0.2</version>
  </dependency>
  <dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-core-asl</artifactId>
    <version>1.9.7</version>
  </dependency>
  <dependency>
    <groupId>com.fasterxml.jackson.datatype</groupId>
    <artifactId>jackson-datatype-json-org</artifactId>
    <version>2.0.2</version>
  </dependency>
  <dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-mapper-asl</artifactId>
    <version>1.9.7</version>
  </dependency>

  <dependency>
    <groupId>commons-digester</groupId>
    <artifactId>commons-digester</artifactId>
    <version>2.1</version>
  </dependency>

  <dependency>
    <groupId>xml-apis</groupId>
    <artifactId>xml-apis</artifactId>
    <version>2.0.2</version>
  </dependency>

  <dependency>
    <groupId>javassist</groupId>
    <artifactId>javassist</artifactId>
    <version>3.4.GA</version>
  </dependency>

  <dependency>
    <groupId>commons-pool</groupId>
    <artifactId>commons-pool</artifactId>
    <version>1.5.4</version>
  </dependency>
  <dependency>
    <groupId>dom4j</groupId>
    <artifactId>dom4j</artifactId>
    <version>1.6.1</version>
  </dependency>

  <dependency>
    <groupId>commons-lang</groupId>
    <artifactId>commons-lang</artifactId>

```

```

    <version>2.1</version>
  </dependency>

  <dependency>
    <groupId>commons-collections</groupId>
    <artifactId>commons-collections</artifactId>
    <version>2.1.1</version>
  </dependency>
  <dependency>
    <groupId>commons-dbc</groupId>
    <artifactId>commons-dbc</artifactId>
    <version>1.3</version>
  </dependency>

  <dependency>
    <groupId>antlr</groupId>
    <artifactId>antlr</artifactId>
    <version>2.7.6</version>
  </dependency>
  <dependency>
    <groupId>aopalliance</groupId>
    <artifactId>aopalliance</artifactId>
    <version>1.0</version>
  </dependency>
  <dependency>
    <groupId>asm</groupId>
    <artifactId>asm</artifactId>
    <version>1.5.3</version>
  </dependency>
  <dependency>
    <groupId>asm</groupId>
    <artifactId>asm-attrs</artifactId>
    <version>1.5.3</version>
  </dependency>
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>1.6.9</version>
  </dependency>
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjtools</artifactId>
    <version>1.6.9</version>
  </dependency>
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>1.6.9</version>
  </dependency>
  <dependency>

```

```

    <groupId>cglib</groupId>
    <artifactId>cglib</artifactId>
    <version>2.1_3</version>
</dependency>
<dependency>
    <groupId>commons-beanutils</groupId>
    <artifactId>commons-beanutils</artifactId>
    <version>1.8.0</version>
</dependency>

<dependency>
    <groupId>net.sf.ehcache</groupId>
    <artifactId>ehcache</artifactId>
    <version>1.2.3</version>
</dependency>

<dependency>
    <groupId>oro</groupId>
    <artifactId>oro</artifactId>
    <version>2.0.8</version>
</dependency>

<dependency>
    <groupId>jaxen</groupId>
    <artifactId>jaxen</artifactId>
    <version>1.1.1</version>
</dependency>

<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbcp2</artifactId>
    <version>2.1</version>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>2.4</version>
            <configuration>
                <archive>
                    <manifest>
                        <addClasspath>true</addClasspath>
                        <classpathPrefix>lib/</classpathPrefix>
                        <mainClass>com.nitinsurana.policereports.GUI</mainClass>
                    </manifest>
                </archive>
            </configuration>
        </plugin>
    </plugins>

```

```

</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.1</version>
  <configuration>
    <source>1.7</source>
    <target>1.7</target>
    <compilerArguments>
      <endorseddirs>${endorsed.dir}</endorseddirs>
    </compilerArguments>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>2.3</version>
  <configuration>
    <failOnMissingWebXml>>false</failOnMissingWebXml>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>2.6</version>
  <executions>
    <execution>
      <phase>validate</phase>
      <goals>
        <goal>copy</goal>
      </goals>
      <configuration>
        <outputDirectory>${endorsed.dir}</outputDirectory>
        <silent>>true</silent>
        <artifactItems>
          <artifactItem>
            <groupId>javax</groupId>
            <artifactId>javaee-endorsed-api</artifactId>
            <version>7.0</version>
            <type>jar</type>
          </artifactItem>
        </artifactItems>
      </configuration>
    </execution>
  </executions>
</plugin>
</plugins>
</build>
</project>

```


ANEXO 10. dispatcher-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:tx="http://www.springframework.org/schema/tx"

       xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd"
">

    <!-- ANOTACIONES -->
    <context:annotation-config />
    <context:component-scan base-package="com.epsgrau.canm" />
    <mvc:annotation-driven />
    <aop:aspectj-autoproxy />

    <!-- RESOLVER LAS VISTAS -->
    <bean id="viewResolver"
          class="org.springframework.web.servlet.view.InternalResourceViewResolver"
          p:prefix="/WEB-INF/jsp/"
          p:suffix=".jsp" />

    <!-- Muestra el index por defecto -->
    <bean name="indexController"
          class="org.springframework.web.servlet.mvc.ParameterizableViewController"
          p:viewName="index" />

    <!-- Identificar llamadas JSON -->
    <bean
class="org.springframework.web.servlet.view.ContentNegotiatingViewResolver">
        <property name="mediaTypes">
            <map>
                <!--<entry key="atom" value="application/atom+xml" />-->
                <!-- El servidor responde en formato json y html-->
                <entry key="htm" value="text/html" />
                <entry key="json" value="application/json" />
            </map>
        </property>
        <property name="defaultViews">
```

```

        <list>
        <bean
class="org.springframework.web.servlet.view.json.MappingJacksonJsonView" />
        </list>
    </property>
</bean>
</beans>

```

ANEXO 11. applicationContext.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"

    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd
    http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd"
">
    <context:annotation-config />
    <context:component-scan base-package="com.epsgrau">
        <context:exclude-filter type="annotation"
expression="org.springframework.stereotype.Controller"/>
    </context:component-scan>

    <bean id="propertyConfigurer"
        class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"
        p:location="/WEB-INF/jdbc.properties"/>

    <!-- EL POOL DE CONEXIONES ES ADMINISTRADO POR C3P0 -->
    <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
destroy-method="close">
        <property name="driverClass" value="{jdbc.driverClassName}" />
        <property name="jdbcUrl" value="{jdbc.url}" />
        <property name="user" value="{jdbc.username}" />
        <property name="password" value="{jdbc.password}" />
        <!-- Numero de conexiones que se abiran cada vez que se requiera -->
        <property name="acquireIncrement" value="1" />
        <!-- Numero de conexiones iniciales -->
        <property name="minPoolSize" value="1" />
        <!-- Conexiones como maximo -->
        <property name="maxPoolSize" value="200" />
        <!-- Maximos statements en cache -->

```

```

    <property name="maxStatements" value="50"/>
    <!-- Maximo tiempo de inactividad -->
    <property name="maxIdleTime" value="10"/>
    <!-- Maximo tiempo de inactividad por las conexiones mayores al pool inicial -->
    <!-- (Si excede el tiempo, el numero de conexiones regresara al pool inicial) -->
    <property name="maxIdleTimeExcessConnections" value="120"/>
    <!-- Hilos multiples (simultaneamente) -->
    <property name="numHelperThreads" value="3"/>
</bean>

<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"
scope="singleton">
    <property name="dataSource" ref="dataSource" />
    <property name="packagesToScan" value="com.epsgrau.canm.entity"/>
    <property name="hibernateProperties">
        <props>
            <!--<prop
key="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</prop-->
            <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
            <!--<prop
key="hibernate.dialect">com.epsgrau.canm.helpers.MyHsqlDialect</prop-->
            <!--<prop
key="hibernate.dialect">org.hibernate.dialect.MySQL5InnoDBDialect</prop-->
            <prop key="hibernate.connection.characterEncoding">UTF-8</prop>
            <prop key="hibernate.format_sql">true</prop>
            <prop key="hibernate.show_sql">true</prop>
        </props>
    </property>
</bean>

<bean id="transactionManager"
class="org.springframework.orm.hibernate3.HibernateTransactionManager">
    <property name="sessionFactory" ref="sessionFactory" />
</bean>
<tx:annotation-driven transaction-manager="transactionManager"/>
</beans>

```

ANEXO 12. spring-security.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:security="http://www.springframework.org/schema/security"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security-3.0.3.xsd">

    <security:http auto-config="true">

```

```

    <security:intercept-url pattern="/BASE-INF/**"
access="IS_AUTHENTICATED_ANONYMOUSLY" />
    <security:intercept-url pattern="/logon.htm"
access="IS_AUTHENTICATED_ANONYMOUSLY" />
    <!-- Las paginas que forman parte de tu aplicacion se puedan visualizar si esta
    logeado como usuario o como administrador -->
    <security:intercept-url pattern="/**" access="ROLE_USER, ROLE_ADMIN" />
    <!--security:form-login authentication-failure-url="/logon.htm?login_error=1"
default-target-url="/home.htm" login-page="/logon.htm" always-use-default-
target="true"/-->
    <security:form-login authentication-failure-url="/logon.htm" default-target-
url="/index.htm" login-page="/logon.htm" always-use-default-target="true"/>
    <security:logout logout-success-url="/logon.htm" />
</security:http>

<security:authentication-manager>
    <security:authentication-provider>
        <security:password-encoder hash="md5"/>
        <security:jdbc-user-service data-source-ref="dataSource"
            users-by-username-query=
                "select username,password, enabled from users where username=?"
            authorities-by-username-query=
                "select username, role from user_roles where username =? " />
        <!-- Sin BD -->
        <!--security:user-service>
            <security:user name="nole" password="123456"
authorities="ROLE_ADMIN"/>
        </security:user-service-->
    </security:authentication-provider>
</security:authentication-manager>
</beans>

```

ANEXO 13. web.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">

    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/applicationContext.xml, /WEB-INF/spring-security.xml
        </param-value>
    </context-param>

    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-
class>

```

```

</listener>
<listener>
  <listener-class>com.epsgrau.canm.helpers.ShutdownListener</listener-class>
</listener>
<listener>
  <listener-
class>org.springframework.web.context.request.RequestContextListener</listener-class>
  </listener>

<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>*.htm</url-pattern>
</servlet-mapping>

<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>

<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
  <init-param>
    <param-name>forceEncoding</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>
<!-- / -> contexto de tu aplicacion, para que me reconosca las extensiones -->
<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>

```

```

    <url-pattern>/*</url-pattern>
</filter-mapping>

<welcome-file-list>
    <welcome-file>logon.htm</welcome-file>
</welcome-file-list>
</web-app>

```

ANEXO 14. jdbc.properties

```

jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/bdsica
jdbc.username=nuestro_usuario
jdbc.password=nuestra_clave

```

ANEXO 15. Log4j.properties

```

# To change this license header, choose License Headers in Project Properties.
# To change this template file, choose Tools | Templates
# and open the template in the editor.
log4j.rootCategory=OFF

```

```

log4j.logger.com.epsgrau.canm.dao=ALL,loggerSicaDao
log4j.appender.loggerSicaDao=org.apache.log4j.FileAppender
log4j.appender.loggerSicaDao.File=${catalina.base}/logs/SicaDao.log
log4j.appender.loggerSicaDao.layout=org.apache.log4j.PatternLayout
log4j.appender.loggerSicaDao.layout.ConversionPattern=%d{ISO8601} %-5p %C %L
[%t] %c : %m %n

```

```

log4j.logger.com.epsgrau.canm.controller=ALL,loggerSicaController
log4j.appender.loggerSicaController=org.apache.log4j.FileAppender
log4j.appender.loggerSicaController.File=${catalina.base}/logs/SicaController.log
log4j.appender.loggerSicaController.layout=org.apache.log4j.PatternLayout
log4j.appender.loggerSicaController.layout.ConversionPattern=%d{ISO8601} %-5p %C
%L [%t] %c : %m %n

```

ANEXO 16. ProgramacionPagoController.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

package com.epsgrau.canm.controller;

```

```

import com.epsgrau.canm.dto.AcreenciaProgramacionDTO;
import com.epsgrau.canm.dto.AcreenciaProgramacionPendienteDTO;
import com.epsgrau.canm.helpers.HelperMensajes;
import com.epsgrau.canm.service.ProgramacionPagoService;
import com.epsgrau.canm.dto.MensajeDTO;
import com.epsgrau.canm.service.AcreenciaService;

```

```

import java.security.Principal;
import java.util.ArrayList;
import java.util.List;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

/**
 *
 * @author Carnol
 */
@Controller
public class ProgramacionPagoController {

    @Autowired
    private ProgramacionPagoService pps;

    @Autowired
    private AcreenciaService as;

    @Autowired
    private HelperMensajes mensajes;

    @Autowired
    private MensajeDTO mensaje;

    private static org.slf4j.Logger loggerController =
    LoggerFactory.getLogger(ProgramacionPagoController.class);

    @RequestMapping(value = "generarProgramacionPagos.htm", method =
    RequestMethod.POST)
    public @ResponseBody
    MensajeDTO generarProgramacionPagos(@RequestParam(value = "nroAcreencia",
    required = false, defaultValue = "") Integer nroAcreencia,
    @RequestParam(value = "importe", required = false, defaultValue = "") Double
    importe,
    @RequestParam(value = "nroCuotas", required = false, defaultValue = "") Integer
    nroCuotas,
    @RequestParam(value = "desde", required = false, defaultValue = "") String
    txtFechDesd,
    @RequestParam(value = "referencia", required = false, defaultValue = "") String
    referencia,
    Principal principal) throws Exception {
        String user = principal.getName();
        try {

```

```

        mensaje = pps.generarProgramacionPago(nroAcreencia, importe, nroCuotas,
txtFechDesd, referencia, user);
        loggerController.info("generarProgramacionPagos: Proceso ejecutado
correctamente");
    } catch (Exception ex) {
        mensaje.setCodResultado("1");
        mensaje.setMensaje(ex.getMessage());
        loggerController.error(ex.getMessage() + "\tCausa: " + ex.getCause());
    }
    return mensaje;
}

```

```

@RequestMapping(value = "generarReprogramacionPagos.htm", method =
RequestMethod.POST)
public @ResponseBody
MensajeDTO generarReprogramacionPagos(@RequestParam(value = "nroAcreencia",
required = false, defaultValue = "") Integer nroAcreencia,
    @RequestParam(value = "importe", required = false, defaultValue = "") Double
importe,
    @RequestParam(value = "nroCuotas", required = false, defaultValue = "") Integer
nroCuotas,
    @RequestParam(value = "desde", required = false, defaultValue = "") String
txtFechDesd,
    @RequestParam(value = "referencia", required = false, defaultValue = "") String
referencia,
    Principal principal) throws Exception {
    String user = principal.getName();
    try {
        mensaje = pps.generarReprogramacionPago(nroAcreencia, importe, nroCuotas,
txtFechDesd, referencia, user);
        loggerController.info("generarProgramacionPagos: Proceso ejecutado
correctamente");
    } catch (Exception ex) {
        mensaje.setCodResultado("1");
        mensaje.setMensaje(ex.getMessage());
        loggerController.error(ex.getMessage() + "\tCausa: " + ex.getCause());
    }
    return mensaje;
}

```

```

@RequestMapping(value = "obtenerAcreenciaProgramacion.htm", method =
RequestMethod.POST)
public @ResponseBody
List<AcreenciaProgramacionDTO>
obtenerAcreenciaProgramacion(@RequestParam(value = "id", required = false,
defaultValue = "") Integer id) {
    try {
        loggerController.info("obtenerAcreenciaProgramacion: Proceso ejecutado
correctamente");
        return as.obtenerAcreenciaProgramacionPorId(id);
    }
}

```



```

    } catch (Exception e) {
        loggerController.error(e.getMessage() + "\tCausa: " + e.getCause());
        return new ArrayList<AcreenciaProgramacionDTO>();
    }
}

@RequestMapping(value = "obtenerDatosAcreenciaProgramacionPendiente.htm",
method = RequestMethod.POST)
public @ResponseBody
List<AcreenciaProgramacionPendienteDTO>
obtenerDatosAcreenciaProgramacionPendiente(@RequestParam(value = "id", required =
false, defaultValue = "") Integer id) {
    try {
        loggerController.info("obtenerDatosAcreenciaProgramacionPendiente: Proceso
ejecutado correctamente");
        return as.obtenerAcreenciaProgramacionPendientePorId(id);
    } catch (Exception e) {
        loggerController.error(e.getMessage() + "\tCausa: " + e.getCause());
        return new ArrayList<AcreenciaProgramacionPendienteDTO>();
    }
}
}
}

```

ANEXO 17. PagoController.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.epsgrau.canm.controller;

import com.epsgrau.canm.helpers.HelperMensajes;
import com.epsgrau.canm.service.PagoService;
import com.epsgrau.canm.dto.MensajeDTO;
import com.epsgrau.canm.dto.ProgramacionPagoAcreedorDTO;
import com.epsgrau.canm.entity.Acreedor;
import com.epsgrau.canm.entity.Asientocontable;
import com.epsgrau.canm.entity.Asientocontabledetalle;
import com.epsgrau.canm.entity.Beneficiario;
import com.epsgrau.canm.entity.Pago;
import com.epsgrau.canm.entity.Programacionpago;
import com.epsgrau.canm.entity.Tipopago;
import com.epsgrau.canm.helpers.Funciones;
import com.epsgrau.canm.service.AsientoContableDetalleService;
import com.epsgrau.canm.service.AsientoContableService;
import java.security.Principal;
import java.util.ArrayList;
import java.util.Date;

```

```

import java.util.List;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

/**
 *
 * @author Carnol
 */
@Controller
public class PagoController {

    @Autowired
    private PagoService ps;

    @Autowired
    private AsientoContableService acs;

    @Autowired
    private AsientoContableDetalleService acds;

    @Autowired
    private HelperMensajes mensajes;

    @Autowired
    private MensajeDTO mensaje;

    private static org.slf4j.Logger loggerController =
    LoggerFactory.getLogger(PagoController.class);

    @RequestMapping(value = "agregarPagoSemestral.htm", method =
    RequestMethod.POST)
    public @ResponseBody
    MensajeDTO guardarPagoSemestral(@RequestParam(value = "idProgPag", required =
    false, defaultValue = "") Integer idProgPag,
    @RequestParam(value = "idAcreeedor", required = false, defaultValue = "") Integer
    idAcreeedor,
    @RequestParam(value = "idBeneficiario", required = false, defaultValue = "")
    Integer idBeneficiario,
    @RequestParam(value = "fechaPago", required = false, defaultValue = "") String
    fechaPago,
    @RequestParam(value = "monto", required = false, defaultValue = "") Double
    monto,
    @RequestParam(value = "referencia", required = false, defaultValue = "") String
    referencia,

```

```

        @RequestParam(value = "tipoPago", required = false, defaultValue = "") Integer
tipoPago,
        @RequestParam(value = "fecha", required = false, defaultValue = "") String fecha,
Principal principal) throws Exception {
    try {
        String user = principal.getName();
        Pago p = new Pago();
        p.setIdProgramacionPago(new Programacionpago(idProgPag));
        p.setIdAcreedor(new Acreedor(idAcreedor));
        p.setIdBeneficiario(new Beneficiario(idBeneficiario));
        Date datePago = Funciones.obtenerFecha(fechaPago);
        p.setFechaPago(datePago);
        p.setMonto(monto);
        p.setReferencia(referencia);
        p.setIdTipoPago(new Tipopago(tipoPago));
        p.setIndActivo("S");
        p.setUserAdd(user);
        Date dateAdd = Funciones.obtenerFechaHora(fecha);
        p.setDateAdd(dateAdd);

        Asientocontable ac = new Asientocontable();
        ac.setFechaAsiento(dateAdd);
        ac.setGlosa("Glosa");
        ac.setDocumento("Documento");
        ac.setObservacion("Obsv");
        ac.setEstado("1");
        ac.setUserAdd(user);
        ac.setDateAdd(dateAdd);
        ac.setIndActivo("S");

        Integer idAsientCont = acs.obtenerUltimoID();
        System.out.println("idAsientoContable: " + idAsientCont);
        Asientocontabledetalle acd = new Asientocontabledetalle();
        acd.setNroCuenta("1");
        acd.setImporte(monto);
        acd.setUserAdd(user);
        acd.setDateAdd(dateAdd);
        acd.setIndActivo("S");

        mensaje = ps.guardar(p, ac, acd);
        if (mensaje.getCodResultado().equals("0")) {
            loggerController.info("guardarPagoSemestral: Proceso ejecutado
correctamente");
        } else {
            loggerController.error("ERROR: " + mensaje.getMensaje());
        }
    } catch (Exception ex) {
        mensaje.setCodResultado("1");
        mensaje.setMensaje(ex.getMessage());
        loggerController.error(ex.getMessage() + "\tCausa: " + ex.getCause());
    }
}

```

```

    }
    return mensaje;
}

@RequestMapping(value = "obtenerProgPagoAcreed.htm", method =
RequestMethod.POST)
public @ResponseBody
List<ProgramacionPagoAcreedorDTO> obtenerProgPagoAcreed(@RequestParam(value
= "nroAcreencia", required = false, defaultValue = "") Integer nroAcreencia,
    @RequestParam(value = "annoMes", required = false, defaultValue = "") String
annoMes) {
    try {
        loggerController.info("obtenerProgPagoAcreed: Proceso ejecutado
correctamente");
        return ps.obtenerProgramacionPagoAcreedor(nroAcreencia, annoMes);
    } catch (Exception e) {
        System.out.println("error controller");
        loggerController.error(e.getMessage() + "\nCausa: " + e.getCause());
        return new ArrayList<ProgramacionPagoAcreedorDTO>();
    }
}

@RequestMapping(value = "verificarPago.htm", method = RequestMethod.POST)
public @ResponseBody
MensajeDTO verificarPagoCuota(@RequestParam(value = "nroAcreencia", required =
false, defaultValue = "") Integer nroAcreencia,
    @RequestParam(value = "annoMes", required = false, defaultValue = "") String
annoMes) {
    try {
        mensaje = ps.verificarPagoCuota(nroAcreencia, annoMes);
        loggerController.info("verificarPagoCuota: Proceso ejecutado correctamente");
    } catch (Exception e) {
        mensaje.setCodResultado("1");
        mensaje.setMensaje(e.getMessage());
        loggerController.error(e.getMessage() + "\nCausa: " + e.getCause());
    }
    return mensaje;
}

@RequestMapping(value = "procesoExtornarPagos.htm", method =
RequestMethod.POST)
public @ResponseBody
MensajeDTO procesoExtornarPagos(@RequestParam(value = "nroAcreencia", required
= false, defaultValue = "") Integer nroAcreencia,
    @RequestParam(value = "annoMes", required = false, defaultValue = "") String
annoMes,
    @RequestParam(value = "referencia", required = false, defaultValue = "") String
referencia,
    Principal principal) {
    try {

```

```

String user = principal.getName();
mensaje = ps.extornarPagos(nroAcreencia, annoMes, referencia, user);
loggerController.info("verificarPagoCuota: Proceso ejecutado correctamente");
} catch (Exception e) {
    mensaje.setCodResultado("1");
    mensaje.setMensaje(e.getMessage());
    loggerController.error(e.getMessage() + "\tCausa: " + e.getCause());
}
return mensaje;
}
}

```

ANEXO 18. GUÍA DE OBSERVACIÓN NRO. 1 (INDICADOR 01)

Tiempo de generación de reportes por los usuarios del área de asesoría legal y administración.

Investigación: Sistema de Control de Acreencias

Objetivo: Medir el tiempo promedio de la generación de reportes.

Instrucciones: Medir el tiempo en (ms) desde que se hace click en botón exportar Excel hasta que se obtiene el archivo generado. Se generará de los procesos que se han realizado pruebas.

Datos Generales:

Nº de Técnico asignado para guiar a los usuarios: 1.

Oficina: Asesoría Legal

REPORTE	TIEMPO INICIO (ms)	TIEMPO FIN (ms)	TIEMPO TRANSCURRIDO (ms)
Programación de Pagos	1453349100145	1453349100724	579 = 0 = 1 seg.
Reprogramación de Pagos	1453349154271	1453349154694	423 = 0 = 1seg.
Pagos Realizados Semestral	1453349237263	1453349237625	362 = 0 = 1seg.
TOTAL	4360047491679	4360047493043	1364 = 1.364 = 2 seg.
TIEMPO PROMEDIO	1453349163893	1453349164347.6667	454.6667 = 0 = 1 seg.

Tabla 55. Guía de Observación Nro. 1

Fuente: Elaborado por el Autor

Área: _____ **Cargo:** _____

Observación: _____ **Fecha:** _____

Notas: _____

REVISADO X PROCESADO X ARCHIVADO X

ANEXO 19. GUÍA DE OBSERVACIÓN NRO. 2 (INDICADOR 02)

Tiempo de generación de reportes por los usuarios del área de asesoría legal y administración.

Investigación: Sistema de Control de Acreencias

Objetivo: Medir el número de órdenes de pagos duplicados

Instrucciones: Medir el número de órdenes de pagos duplicados y se contrastará con el número de deudas pendientes.

Datos Generales:

Nº de Técnico asignado para guiar a los usuarios: 1.

Oficina: Administración

FECHA	CONTEO	SUBTOTAL (ÓRDENES DUPLICADOS)	SUBTOTAL (DEUDAS PENDIENTES)
20/04/2015	IIIIII	8	20
21/04/2015	IIII	5	15
22/01/2015	IIIIII	7	27
NÚMERO TOTAL		20	20.6667
NÚMERO PROMEDIO		6.6667 = 7	193.3333 = 194

Tabla 56. Guía de Observacion Nro. 2

Fuente: Elaborado por el Autor

Área: _____ **Cargo:** _____

Observación: _____ **Fecha:** _____

Notas: _____

REVISADO X PROCESADO X ARCHIVADO X

ANEXO 20. ENCUESTA NRO. 1 (INDICADOR 03 – NIVEL OPERACIONAL)

Instrucciones Generales: La presente encuesta está dirigido para los usuarios del Nivel Operacional y que se encuentra en el al área de Asesoría Legal cuyo objetivo es medir el grado de satisfacción de los usuarios.

Debe escribir con una (x) en una de las columnas.

	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
¿El SICA cumple mis especificaciones a nivel operativo?					
¿Estoy satisfecho con el trabajo que SICA realiza?					
¿La información proporcionada fue de utilidad?					
¿Con el SICA, soy capaz de completar mi trabajo?					
¿El SICA me permite llevar a cabo funciones, con fiabilidad?					
¿El output del SICA es fácil de entender?					
¿Es fácil solucionar un error operativo?					

¿La comprobación del SICA,
para asegurarse de que se ejecuta
correctamente, fue fácil?

¿El SICA es fácil de usar?

¿Estoy muy contento con el
SICA?

Tabla 57. Encuesta Nro. 1
Fuente: Elaborado por el Autor

ANEXO 21. ENCUESTA NRO. 2 (INDICADOR 03 – NIVEL ESTRATÉGICO)

Instrucciones Generales: La presente encuesta está dirigida para los usuarios del Nivel Estratégico y que pertenecen al área de Administración cuyo objetivo es medir el grado de satisfacción de los usuarios.

Debe escribir con una (x) en una de las columnas.

	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
¿El SICA cumple mis especificaciones a nivel estratégico?					
¿La información presntada por el SICA es lo que realmente esperaba ver?					

¿La información proporcionada me ayudó para una mejor toma de decisiones?					
¿Con el SICA, soy capaz de completar mi trabajo?					
¿La información presentada por el SICA es clara?					
¿El funcionamiento del SICA es entendible?					
¿El SICA presenta una interfaz amigable?					
¿La comprobación del SICA, para asegurarse de que se ejecuta correctamente, fue fácil?					
¿El SICA es fácil de usar?					
¿Estoy muy contento con el SICA?					

Tabla 58. Encuesta Nro. 2
Fuente: Elaborado por el Autor

ANEXO 22. DIAGRAMA DE CASOS DE USO DEL SISTEMA

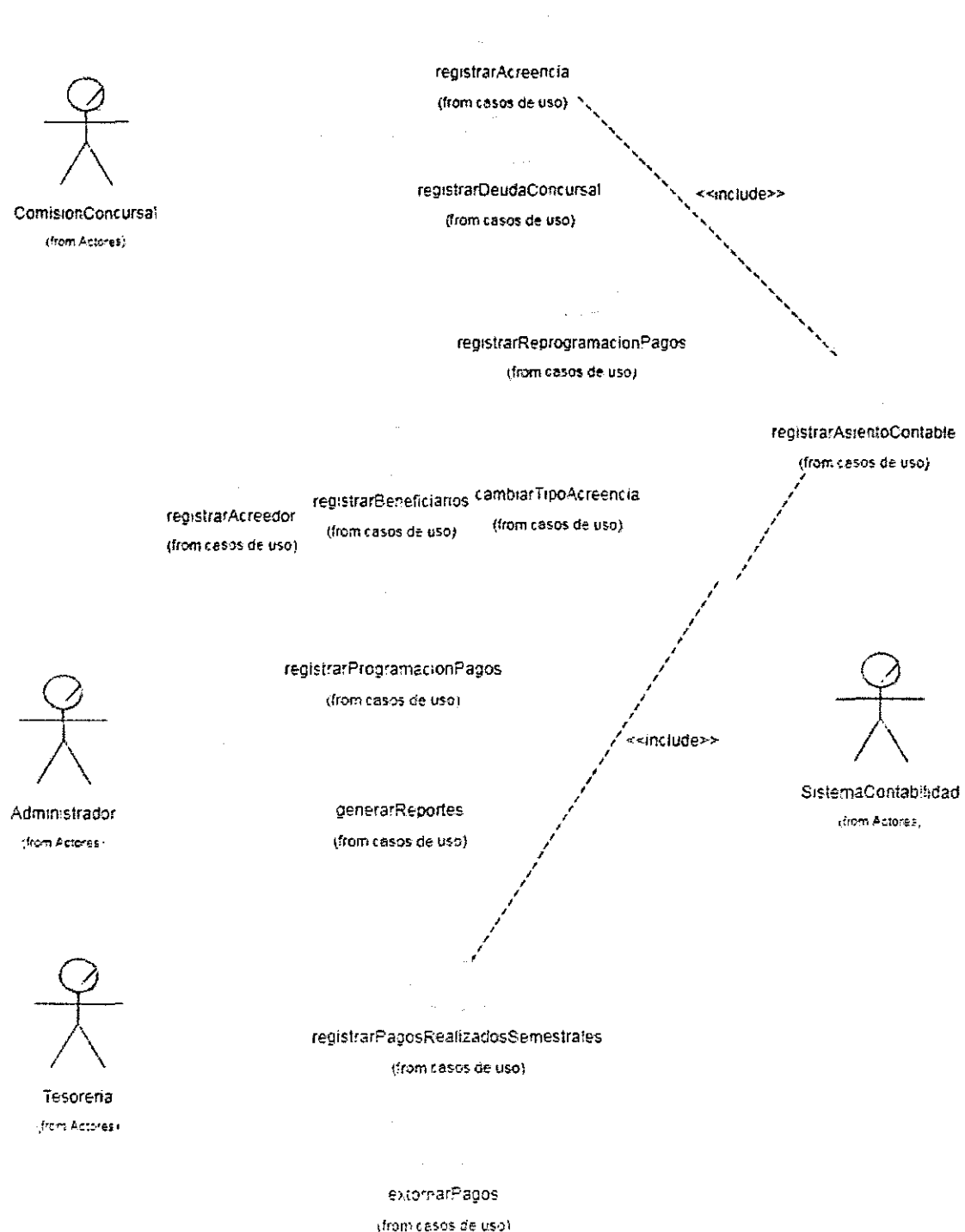


Figura 65. DCU Sistema - SICA
Fuente: Elaborado por el Autor.

ANEXO 23. DIAGRAMA DE CASOS DE USO DEL NEGOCIO

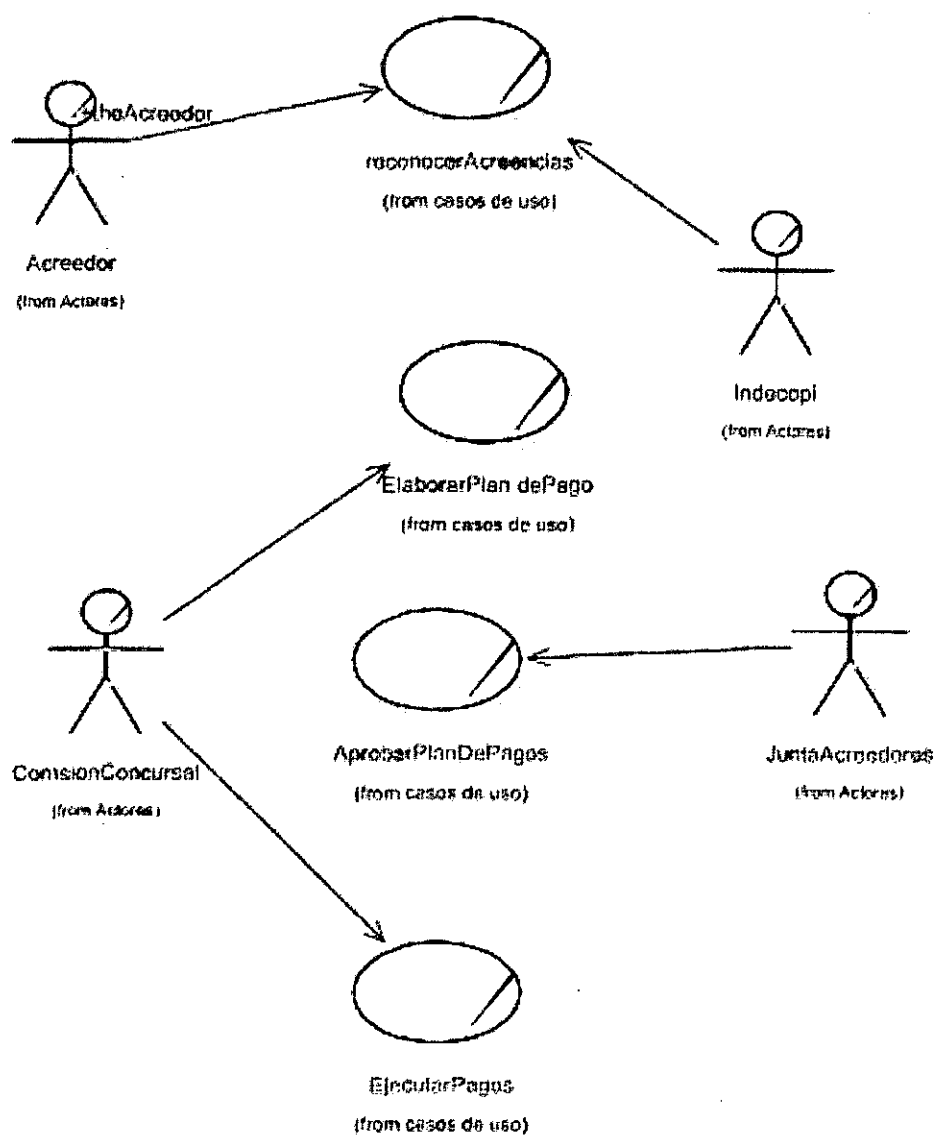
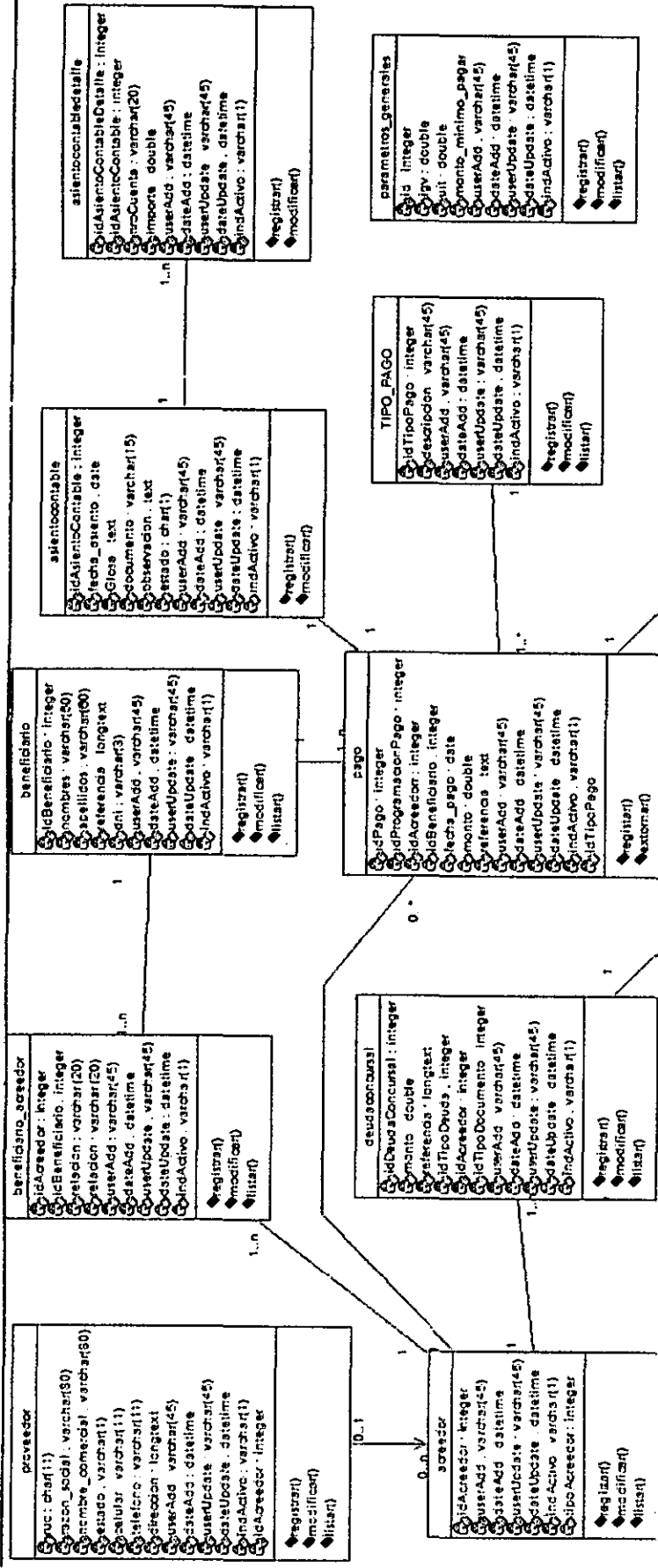


Figura 66. DCU Negocio – SICA
Fuente: Elaborado por el Autor.

ANEXO 24. DIAGRAMA DE CLASES



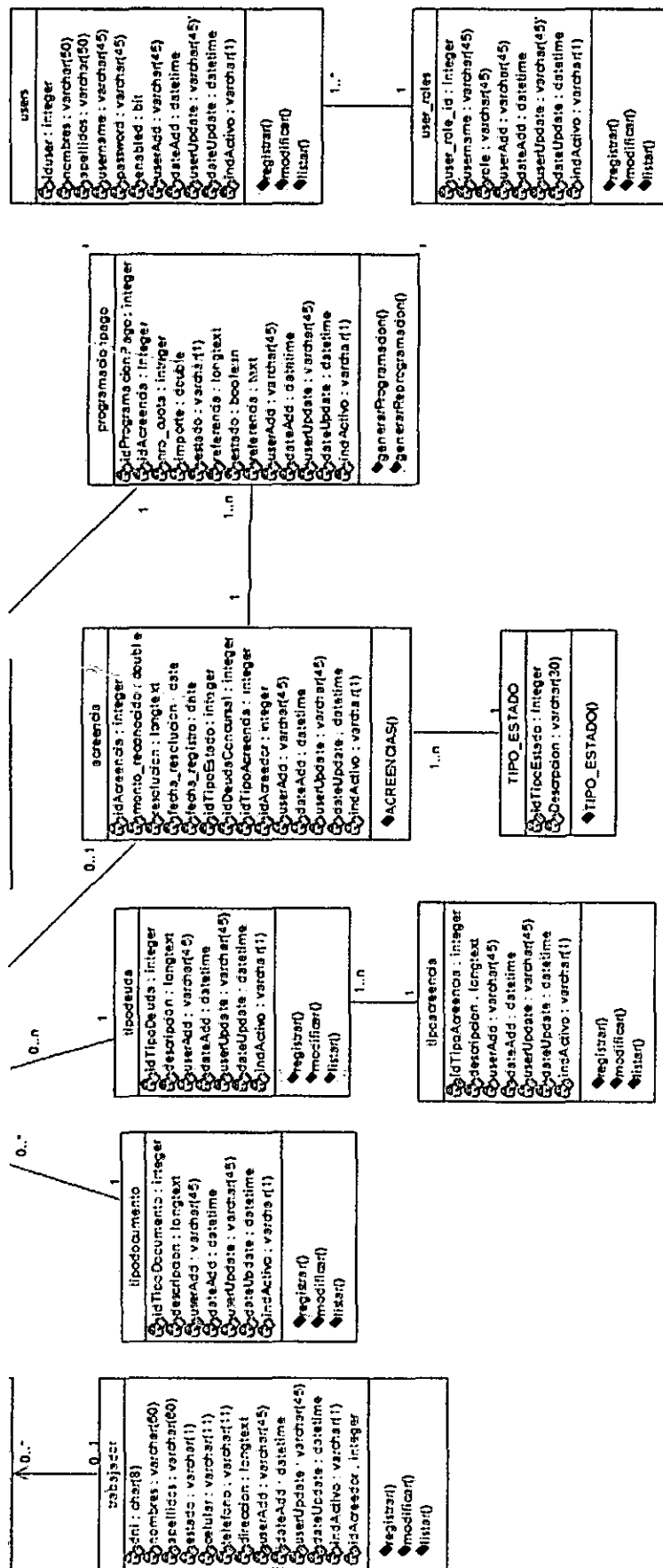


Figura 67. Diagrama de Clases - SICA
Fuente: Elaborado por el Autor.

GLOSARIO DE TÉRMINOS

1. **ACTIVIDAD EMPRESARIAL.-** Actividad económica, habitual y autónoma en la que confluyen los factores de producción, capital y trabajo, desarrollada con el objeto de producir bienes o prestar servicios.
2. **ACREEDOR.-** Persona natural o jurídico, sociedades conyugales, sucesiones indivisas y otros patrimonios autónomos que sean titulares de un crédito.
3. **COMISION.-** La Comisión de Procedimientos Concursales del INDECOPI y las Comisiones que se instalen en virtud de convenios.
4. **CREDITO.-** Derecho del acreedor a obtener una prestación asumida por el deudor como consecuencia de una relación jurídica obligatoria.
5. **DEUDOR.-** Persona natural o jurídico, sociedades conyugales y sucesiones indivisas. Se incluye a las sucursales en el Perú de organizaciones o sociedades extranjeras.
6. **INDECOPI.-** Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual.
7. **JUNTA.-** Junta de Acreedores.
8. **LEY.-** Ley General del Sistema Concursal.
9. **SISTEMA CONCURSAL.-** El Sistema Concursal está conformado por las normas aplicables a los procedimientos concursales, por los agentes que intervienen en los procedimientos concursales, así como por las Autoridades Administrativas y Judiciales a las que la Ley y/o sus normas complementarias o modificatorias asigne competencia.
10. **TUPA.-** Texto Único de Procedimientos Administrativos del INDECOPI.

11. **TRIBUNAL.-** El Tribunal de Defensa de la Competencia y de la Protección de la Propiedad del INDECOPI.
12. **BACK – END.-** Puede traducirse como motor, dorsal final o zaga es la capa de acceso a datos (lado del servidor).
13. **FRONT – END.-** Puede traducirse como Interfaz final o frontal es la capa de presentación (lado del cliente).
14. **AJAX.-** Acrónimo de JavaScript asíncrono y XML es una técnica de desarrollo web para crear aplicaciones interactivas ejecutándose en el cliente mientras se comunican con el servidor en segundo plano.
15. **JSON.-** Acrónimo de JavaScript Object, es un formato ligero para el intercambio de datos.
16. **IDE.-** Se traduce como Ambiente de Desarrollo Integrado es una aplicación informática que proporciona servicios integrales para facilitarle al programador el desarrollo de software.
17. **GUI.-** Se traduce como Interfaz Gráfica de Usuario, utilizando un conjunto de imágenes y objetos gráficos para representar información y acciones disponibles en la interfaz de usuario.
18. **DEBUGGER.-** Se traduce como depurador, es un programa usado para probar y eliminar los errores de otros programas.
19. **POJO.-** Es un instancia de una clase que no extiende ni implementa nada en especial.
20. **IEEE.-** Se traduce como Instituto de Ingeniería Eléctrica y Electrónica, es una asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.
21. **JDK.-** Acrónimo de Java Development Kit, provee herramientas de desarrollo para la creación de programas Java.